Contents




Abstract. This paper has two main results:

(1) Extend results of Reference 2 to allow for bubble wall velocity simulations for a wide range of laser pulse energies ($E$) and laser spot sizes ($d$).
(2) Determined optimal initial and final bubble radius simulation values that have up to about 6 times less error than Figure 9 in Reference 2.

Main result 1 includes an estimator for the proportionality constant $k$ in Eqn. 12 of Reference 2 between a laser's pulse energy and the change in volume of the bubble it generates. This estimator for $k$, $\hat{k}$, is used in the bubble wall velocity simulation to determine the final bubble radius value. Being able to compute the final bubble radius using $\hat{k}$ allows for running the simulation for a wide range of laser pulse energies, spot sizes, and pulse durations. Allowing for the simulation of different lasers is important for the optimization of a communication protocol across the air-water boundary. The authors of Reference 2 can only run simulations for fixed laser pulse energies and spot sizes. Main result 2 uses a Sum of Squared Errors (SSE) approach to find optimal initial and final radius values to fit as close as possible to experimental bubble velocity data (Figure 5 of Reference 2). The SSE between experimental data and a bubble wall velocity simulation using optimal radius values found in Main result 2 is up to 6 times smaller than SSE between the same experimental data and a bubble wall velocity simulation with radius values used by the authors of Reference 2.

The simulation results of Figure 9 in Reference 2 were reproduced in order to assert the physics of bubble wall expansion are understood and implemented correctly.

Lastly, two models are presented

(1) Viscous Wave Equation
(2) Partial Differential Equations (PDEs) governing bubble collapse

Implementation details to simulate the Viscous Wave Equation is detailed in both 1 and 2 dimensions. The PDEs governing bubble collapse are decoupled to isolate a bubble's pressure and temperature, thus leading to an approximation of the bubble wall collapse velocity. Chapter V describes how Chapters II, III, and IV are connected and could help accelerate the construction of an efficient communication protocol across the air-water boundary.

# 1. Chapter I, Introduction

In 1880 Alexander Graham Bell found that a beam of light focused to a point on metals (such as gold, silver, platinum, and iron) can generate sound waves [1]. Since this discovery, many experiments have been conducted with light and a variety of different media. The Naval Research Laboratory (NRL) is interested in learning about acoustic profiles generated in water media upon impact of a laser pulse, and some of their findings are reported in [3]. The goal is to relate properties of visible and near infrared light (such as wavelength, pulse duration, spot size, and energy) to properties of the acoustic wave (such as pressure amplitude, direction, and shock velocity) generated by the laser. Being able to understand the relationship between the two could potentially lead to communication protocols across the air-water boundary with minimal noise. This work focuses on the use of lasers as a form of visible and near infrared light and, since laser equipment can be expensive, presents different types of simulations that can help decide which laser could be procured. Consider the scenario shown in Figure 1. An airborne pulsed laser can be focused on the water surface for a duration of time (usually a nanoseconds or picoseconds). During this short duration of laser emission, the water molecules heat up and form a thin elongated plasma filament under water. This thin filament under water undergoes thermal expansion and forms an acoustic pulse. The acoustic pulse travels a certain distance under water, creating expanding and collapsing pressure bubbles along the way, eventually attenuating to undetectable levels. Ideally, before severe attenuation occurs, an acoustic pulse can be detected by a hydrophone far below the water surface and converted to either a binary 0 or 1, thus potentially leading to a communication protocol. Chapter II details implementing the viscous wave equation in 1 and 2 dimensions. The authors of [2] implement simulations of spherical bubbles generated by picosecond and nanosecond lasers, and a 1-dimensional wave model is applicable to the results of [2] if you assume a pressure bubble has spherical symmetry and no angular dependence (i.e. no matter how you rotate the bubble, it looks the same). A 2-dimensional model is also relevant if you assume the bubble is any object that has an axis of symmetry (i.e. a perfect cylinder or an ellipsoid). The 2-dimensional viscous wave equation model serves as a realistic numerical model to describe spherical and ellipsoidal pressure bubbles resulting from laser pulses with different pulse duration and energy. Chapter III details implementing differential equations governing bubble wall expansion velocity. Since a bubble's expansion velocity is linked to its change in pressure over time, these simulations can be used to generate acoustic pressure profiles as desired. The results in Figure 9 of [2] are reproduced to assert that the physics is implemented correctly. These results are significantly improved upon finding optimal bubble initial and final radius values. Bubble expansion is half the story and bubble collapse is the other half. Chapter IV goes through a detailed derivation of decoupling partial differential equations governing bubble collapse, isolating a bubble's temperature and pressure.

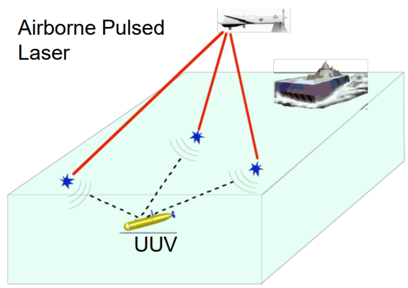



**Figure 1.** This image is taken from [3]

## 2. Chapter II, Viscous Wave Equation

2.1. **1-Dimensional Viscous Wave Equation.** In order to understand how underwater acoustic pressure waves propagate, the viscous wave equation should be numerically solved. The 1-Dimensional Viscous Wave Equation found in Eqn. 16 of [4] is a good start :

$$\frac{4v}{3c_0^2}p_{xxt} + p_{xx} - \frac{1}{c_0^2}p_{tt} = 0, \tag{2.1}$$

where $p$ is pressure, $v = \frac{\mu}{\rho}$, the kinematic viscosity coefficient and $\mu$ and $\rho$ are the bulk viscosity coefficient and initial density. The subscripts refer to differentiation with respect to space ($x$) and time ($t$). $c_0 \approx 1480\frac{m}{s}$ is the speed of sound in water. 2.1 is derived in Eqns. 1 and 2 in reference [4] (conservation of momentum, mass, and equation of state). It is important to note that 2.1 is approximated linearly in [4] using Taylor Series. This means quadratic or higher order fluctuations in density are not used. The rest of this section goes into the details of how this equation can be numerically solved.

Let $dx$ be a small spatial step and $dt$ be a small temporal step. Let $n$ represent the $n^{th}$ time step. Let $i$ represent the $i^{th}$ spatial step. The 3 terms in the equation above $p_{xxt}, p_{xx}, p_{tt}$ can be approximated with finite differences. First, the $p_{xx}$ term, is approximated using central difference:

$$p_{xx}(i,n) = p_{xx} \approx \frac{p_{i+1}^n - 2p_i^n + p_{i-1}^n}{dx^2} \tag{2.2}$$

where the $i$ and $n$ are dropped for clarity. Observe Eqn. 2.1 has a $p_{xxt}$ term. In order to approximate this term, it is assumed $p_t(x_i) = p_t \approx \frac{(p_i^n - p_i^{n-1})}{dt}$ and take a time derivative of Eqn. 2.2. Note that $x_i$ is dropped for clarity. This gives:

$$p_{xxt} \approx \frac{\frac{p_{i+1}^n - p_{i+1}^{n-1}}{dt} - 2\frac{p_i^n - p_i^{n-1}}{dt} + \frac{p_{i-1}^n - p_{i-1}^{n-1}}{dt}}{dx^2} \tag{2.3}$$

Then $p_{tt}$ is approximated:

$$p_{tt} \approx \frac{p_i^{n+1} - 2p_i^n + p_i^{n-1}}{dt^2} \tag{2.4}$$

Plugging in Eqn. 2.4 into 2.1 and solving for $p_{tt}$ gives:

$$\frac{p_i^{n+1} - 2p_i^n + p_i^{n-1}}{dt^2} \approx p_{tt} = \frac{4}{3}vp_{xxt} + c_0^2 p_{xx} \tag{2.5}$$

Finally, solving Eqn. 2.5 for $p_i^{n+1}$, we have:

$$p_i^{n+1} \approx (dt)^2\left(\frac{4}{3}vp_{xxt} + c_0^2 p_{xx}\right) + 2p_i^n - p_i^{n-1} \tag{2.6}$$

Eqn. 2.6 can be implemented on a computer to simulate an acoustic pressure wave. Let $\Delta t \approx dt$ and $\Delta x \approx dx$. Then Eqn. 2.6 can be rewritten as:

$$p_i^{n+1} \approx 2p_i^n - p_i^{n-1} + c_0^2 p_{xx}(\Delta t)^2 + \frac{4}{3} v p_{xxt}(\Delta t)^2 \tag{2.7}$$

The terms $2p_i^n - p_i^{n-1}$, $\frac{4}{3} v p_{xxt}(\Delta t)^2$, and $c_0^2 p_{xx}(\Delta t)^2$ in Eqn. 2.7 must be converted into matrices $\mathbf{D}_A, \mathbf{D}_B, \mathbf{D}_C \in \mathbb{R}^{2N\text{x}2N}$ respectively, where $N$ is the total number of spatial data points. This leads to a final transformation matrix defined $\mathbf{D} = \mathbf{D}_A + \mathbf{D}_B + \mathbf{D}_C \in \mathbb{R}^{2N\text{x}2N}$. Translating Eqn. 2.7 into matrix form is needed to overcome numerical instability. In particular, all of $\mathbf{D}$'s eigenvalues must be less than 1 in magnitude to ensure a numerically stable solution [5].

Next, we convert the terms $2p_i^n - p_i^{n-1}$, $c_0^2 p_{xx}(\Delta t)^2$, and $\frac{4}{3} v p_{xxt}(\Delta t)^2$ in Eqn. 2.7 into matrix form.

Define $\mathbf{D}_A, \mathbf{D}_B$, and $\mathbf{D}_C$ such that

$$\mathbf{D}_A = \left[\begin{array}{cccccccccccc}
0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & \dots & \dots & \dots & 0 \\
0 & 2 & 0 & \dots & \dots & \vdots & 0 & -1 & 0 & \dots & \dots & \vdots \\
\vdots & 0 & 2 & 0 & \dots & \vdots & \vdots & 0 & -1 & 0 & \dots & \vdots \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots & \vdots & \vdots & 0 & \ddots & 0 & \vdots \\
\vdots & \vdots & \vdots & 0 & 2 & 0 & \vdots & \vdots & \vdots & 0 & -1 & 0 \\
0 & \dots & \dots & \dots & 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 \\
 & & & & & & & & & & & \\
\multicolumn{6}{c}{\mathbf{I}^{N\text{x}N}} & \multicolumn{6}{c}{\mathbf{0}^{N\text{x}N}} \\
 & & & & & & & & & & &
\end{array}\right] \tag{2.8}$$

$$\mathbf{D}_B = \left[\frac{c_0 \Delta t}{\Delta x}\right]^2 \begin{bmatrix} \begin{matrix} 0 & 0 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & \vdots \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & \dots & 0 & 0 & 0 \end{matrix} & \mathbf{0}^{N\text{x}N} \\ \mathbf{0}^{N\text{x}N} & \mathbf{0}^{N\text{x}N} \end{bmatrix} \tag{2.9}$$

$$\mathbf{D}_C = \frac{4v\Delta t}{3(\Delta x)^2} \begin{bmatrix} \begin{matrix} 0 & 0 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & \vdots \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & 1 & -2 & 1 \\ 0 & \dots & \dots & 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & 0 & 0 \end{matrix} \\ \mathbf{0}^{N\text{x}N} & \mathbf{0}^{N\text{x}N} \end{bmatrix} \tag{2.10}$$

Let $\mathbf{P} \in \mathbb{R}^{2N\text{x}1}$ represent pressure values.

Define:

$$\mathbf{P} = \begin{bmatrix} p_0^n \\ p_1^n \\ \vdots \\ p_{N-1}^n \\ p_0^{n-1} \\ p_1^{n-1} \\ \vdots \\ p_{N-1}^{n-1} \end{bmatrix} \tag{2.11}$$

Where the subscript represents spatial index and superscript represents temporal index. Note that the size of $\mathbf{P}$ is $2N \times 1$ because the first $N$ elements represent the current time step's pressure values, whereas the last $N$ elements represent the previous time step's pressure values. This forces $\mathbf{D}$ to be $2N \times 2N$. If you initialize $\mathbf{P}$ with a wave-like shape (like a Gaussian) and apply $\mathbf{P} = \mathbf{DP}$, you will have the next time step pressure values. Doing this repeatedly for a finite number of time steps is the main idea of Algorithm 1.

**Algorithm 1** Simulate Viscous Wave Equation

1: **procedure** SIMULATEVWE($\Delta x, \Delta t, v, c_0$)
2:     $wave \leftarrow e^{-\frac{x^2}{10}}$ ▷ Define a Gaussian wave. Can define a different initial shape
3:     $\mathbf{P} \leftarrow [\mathbf{00}]$ ▷ Pressure starts out 0 everywhere, but *wave* is added later
4:     $\mathbf{D} = \mathbf{D}_A + \mathbf{D}_B + \mathbf{D}_C$ ▷ Define $\mathbf{D}$ using Eqns. 2.8-2.10
5:     **for** $t = 1 \longrightarrow 150\Delta t$ **do** ▷ Run for 150 nanoseconds
6:         $\mathbf{P} = \mathbf{DP}$ ▷ Compute next time step's pressure values
7:     **end for**
8: **end procedure**

Algorithm 2 ensures line 6 in Algorithm 1 is numerically stable. Algorithm 2 must be implemented to find values of $\Delta x$ and $\Delta t$ such that $\mathbf{D}$ is numerically stable. This process involves computing $\mathbf{D}$ for a range of $\Delta x$ and $\Delta t$ values such that all of $\mathbf{D}$'s eigenvalues are smaller than 1 in magnitude [5].

Note that the matrices $\mathbf{D}_A, \mathbf{D}_B$, and $\mathbf{D}_C$ are sparse and can be optimized to use less memory on a computer.

**Algorithm 2** Find $dx$ and $dt$ such that all eigenvalues of $\mathbf{D}$ are less than 1

**procedure** FINDDELTAXDELTAT
    **for** $dt = 1 \times 10^{-10} \longrightarrow 1 \times 10^{-9}$ **do**
        **for** $dx = 1 \times 10^{-7} \longrightarrow 1 \times 10^{-6}$ **do**
            $\mathbf{D} \leftarrow \mathbf{D_A} + \mathbf{D_B} + \mathbf{D_C}$ ▷ using Eqns. 2.8-2.10
            $\mathbf{E} = eig(\mathbf{D})$ ▷ Compute all eigenvalues of D
            $M = max(abs(\mathrm{E}))$ ▷ Determine the eigenvalue with maximum magnitude
            **if** $M < 1$ **then**
                **return** $dx$, $dt$, $M$ ▷ numerically stable $dx$ and $dt$
            **end if**
        **end for**
    **end for**
**end procedure**

2.2. **2-Dimensional Viscous Wave Equation.**
Applying the Laplacian ($\nabla^2$ operator) in two dimensions in Eqn. 16 of [4] we have:

$$\frac{4v}{3c_0^2}(P_{xxt}+P_{yyt})+(P_{xx}+P_{yy})-\frac{1}{c_0^2}P_{tt}=0 \tag{2.12}$$

where $x$ and $y$ are spatial dimensions and $t$ is temporal.

Let $i \in [0, I-1]$ represent mesh in $x$-direction.
Let $j \in [0, J-1]$ represent mesh in $y$-direction.

Just as we did in the 1-dimensional case, we use Finite Difference Method to approximate the following terms in Eqn. 2.12:

$$p_{xx} \approx \frac{p^n_{i+1,j}-2p^n_{i,j}+p^n_{i-1,j}}{dx^2} \tag{2.13}$$

$$p_{yy} \approx \frac{p^n_{i,j+1}-2p^n_{i,j}+p^n_{i,j-1}}{dy^2} \tag{2.14}$$

$$p_{xxt} \approx \frac{\frac{p^n_{i+1,j}-p^{n-1}_{i+1,j}}{dt}-2\frac{p^n_{i,j}-p^{n-1}_{i,j}}{dt}+\frac{p^n_{i-1,j}-p^{n-1}_{i-1,j}}{dt}}{dx^2} \tag{2.15}$$

$$p_{yyt} \approx \frac{\frac{p^n_{i,j+1}-p^{n-1}_{i,j+1}}{dt}-2\frac{p^n_{i,j}-p^{n-1}_{i,j}}{dt}+\frac{p^n_{i,j-1}-p^{n-1}_{i,j-1}}{dt}}{dy^2} \tag{2.16}$$

$$p_{tt} \approx \frac{p^{n+1}_{i,j}-2p^n_{i,j}+p^{n-1}_{i,j}}{dt^2} \tag{2.17}$$

Plugging in Eqn. 2.17 into Eqn. 2.12 and solving for $p_{tt}$, we have:

$$\frac{4v}{3}(p_{xxt}+p_{yyt})+c_0^2(p_{xx}+p_{yy})=p_{tt}\approx\frac{p^{n+1}_{i,j}-2p^n_{i,j}+p^{n-1}_{i,j}}{dt^2} \tag{2.18}$$

The next two steps solve for $p^{n+1}_{i,j}$, the future time step's pressure value at the $i^{th}$ and $j^{th}$ position in the $x$ and $y$ mesh respectively.

$$\frac{4v}{3}dt^2(p_{xxt}+p_{yyt})+c_0^2(dt^2)(p_{xx}+p_{yy})=p_{tt}\approx p^{n+1}_{i,j}-2p^n_{i,j}+p^{n-1}_{i,j} \tag{2.19}$$

$$\frac{4v}{3}dt^2(p_{xxt} + p_{yyt}) + c_0^2(dt^2)(p_{xx} + p_{yy}) + 2p_{i,j}^n - p_{i,j}^{n-1} = p_{tt} \approx p_{i,j}^{n+1} \tag{2.20}$$

Just as in the previous section, Eqn. 2.20 needs to be translated into matrix form in order to overcome numerical instability. Let $\mathbf{D}_1, \mathbf{D}_{2A}, \mathbf{D}_{2B}, \mathbf{D}_{3A}, \mathbf{D}_{3B}, \mathbf{D} \in \mathbb{R}^{2IJ \times 2IJ}$ represent each of the 5 terms to the left of Eqn. 2.20, and define $\mathbf{D}$ using Algorithm 3 below. Note that $\mathbf{D}$ is sparse and should be optimized to use less memory on a computer.

**Algorithm 3** Define transformation matrix $\mathbf{D}$

```
procedure DEFINETRANSFORMATIONMATRIX(dx, dy, dt, I, J, c_0, v)
    N = IJ
    const_{D_{2A}} = c_0(dt)^2 / dx
    const_{D_{2B}} = c_0(dt)^2 / dy
    const_{D_{3A}} = 4v(dt) / dx^2
    const_{D_{3B}} = 4v(dt) / dy^2
    for i = N + 1 ⟶ 2N do                                  ▷ Set the identity block matrix
        for j = 1 ⟶ N do
            D(i, j) = 1
        end for
    end for
    for i = 1 ⟶ I − 2 do
        for j = 1 ⟶ J − 2 do
            ij1 = iJ + j + 1                                   ▷ define D_1
            ij2 = iJ + j + 1 + IJ
            D(ij1, ij1) = D(ij1, ij1) + 2
            D(ij1, ij2) = D(ij1, ij2) − 1
            ij2 = (i − 1)J + j + 1                             ▷ Define D_{2A}
            ij3 = iJ + j + 1
            ij4 = (i + 1)J + j + 1
            D(ij1, ij2) = D(ij1, ij2) + const_{D_{2A}}
            D(ij1, ij3) = D(ij1, ij3) − 2const_{D_{2A}}
            D(ij1, ij4) = D(ij1, ij4) + const_{D_{2A}}
            ij5 = iJ + (j − 1) + 1                             ▷ Define D_{2B}
            D(ij1, ij5) = D(ij1, ij5) + const_{D_{2B}}
            D(ij1, ij5 + 1) = D(ij1, ij5 + 1) − 2const_{D_{2B}}
            D(ij1, ij5 + 2) = D(ij1, ij5 + 2) + const_{D_{2B}}
            if ij4 + N ≤ 2N then                               ▷ Define D_{3A}
                D(ij1, ij2) = D(ij1, ij2) + const_{D_{3A}}
                D(ij1, ij3) = D(ij1, ij3) − 2const_{D_{3A}}
                D(ij1, ij4) = D(ij1, ij4) + const_{D_{3A}}
                D(ij1, ij2 + N) = D(ij1, ij2 + N) − const_{D_{3A}}
                D(ij1, ij3 + N) = D(ij1, ij3 + N) + 2const_{D_{3A}}
                D(ij1, ij4 + N) = D(ij1, ij4 + N) − const_{D_{3A}}
            end if
            if ij5 + 2 + N ≤ 2N then                           ▷ Define D_{3B}
                D(ij1, ij5) = D(ij1, ij5) + const_{D_{3B}}
                D(ij1, ij5 + 1) = D(ij1, ij5 + 1) − 2const_{D_{3B}}
                D(ij1, ij5 + 2) = D(ij1, ij5 + 2) + const_{D_{3B}}
                D(ij1, ij5 + N) = D(ij1, ij5 + N) − const_{D_{3B}}
                D(ij1, ij5 + 1 + N) = D(ij1, ij5 + 1 + N) + 2const_{D_{3B}}
                D(ij1, ij5 + 2 + N) = D(ij1, ij5 + 2 + N) − const_{D_{3B}}
            end if
        end for
    end for
end procedure
```

Eqn. 2.11 in the previous section turns into a 2-dimensional matrix of pressure values in this section. Just as before, we need to keep track of the current time step's pressure matrix and the previous time step's pressure matrix.

Let $\mathbf{p}^n, \mathbf{p}^{n-1} \in \mathbb{R}^{I\times J}$.

Let $\mathbf{p} \in \mathbb{R}^{2I\times J}$, and define:

$$\mathbf{p}^n = \begin{array}{cc} \overset{J}{\longleftrightarrow} & \\ \underbrace{\begin{bmatrix} p^n_{0,0} & p^n_{0,1} & p^n_{0,2} & \cdots & p^n_{0,J-1} \\ p^n_{1,0} & p^n_{1,1} & p^n_{1,2} & \cdots & p^n_{1,J-1} \\ p^n_{2,0} & p^n_{2,1} & p^n_{2,2} & \cdots & p^n_{2,J-1} \\ p^n_{0,0} & p^n_{0,1} & p^n_{0,2} & \cdots & p^n_{0,J-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p^n_{I-1,0} & p^n_{I-1,1} & p^n_{I-1,2} & \cdots & p^n_{I-1,J-1} \end{bmatrix}}_{\text{curr. state}} & \updownarrow I \end{array} \tag{2.21}$$

$$\mathbf{p}^{n-1} = \begin{array}{cc} \overset{J}{\longleftrightarrow} & \\ \underbrace{\begin{bmatrix} p^{n-1}_{0,0} & p^{n-1}_{0,1} & p^{n-1}_{0,2} & \cdots & p^{n-1}_{0,J-1} \\ p^{n-1}_{1,0} & p^{n-1}_{1,1} & p^{n-1}_{1,2} & \cdots & p^{n-1}_{1,J-1} \\ p^{n-1}_{2,0} & p^{n-1}_{2,1} & p^{n-1}_{2,2} & \cdots & p^{n-1}_{2,J-1} \\ p^{n-1}_{0,0} & p^{n-1}_{0,1} & p^{n-1}_{0,2} & \cdots & p^{n-1}_{0,J-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p^{n-1}_{I-1,0} & p^{n-1}_{I-1,1} & p^{n-1}_{I-1,2} & \cdots & p^{n-1}_{I-1,J-1} \end{bmatrix}}_{\text{prev state}} & \updownarrow I \end{array} \tag{2.22}$$

$$\mathbf{p} = \begin{array}{cc} \overset{J}{\longleftrightarrow} & \\ \begin{bmatrix} \mathbf{p}^n \\ \mathbf{p}^{n-1} \end{bmatrix} & \updownarrow 2I \end{array} \tag{2.23}$$

Without loss of generality, we can flatten the 2D data structure $\mathbf{p}$ by doing a row scan (you could alternatively do a column scan).

$$\mathbf{p}^n = \left[\begin{array}{c} p^n_{0,0} \\ p^n_{0,1} \\ p^n_{0,2} \\ \vdots \\ p^n_{0,J-1} \\ p^n_{1,0} \\ p^n_{1,1} \\ p^n_{1,2} \\ \vdots \\ p^n_{1,J-1} \\ \vdots \\ p^n_{I-1,0} \\ p^n_{I-1,1} \\ \vdots \\ p^n_{I-1,J-1} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c} p \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow 1^{st} \text{ row} \\ \left.\vphantom{\begin{array}{c} p \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow 2^{nd} \text{ row} \\ \left.\vphantom{\begin{array}{c} \vdots \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow I^{th} \text{ row} \end{array} \quad \mathbf{p}^{n-1} = \left[\begin{array}{c} p^{n-1}_{0,0} \\ p^{n-1}_{0,1} \\ p^{n-1}_{0,2} \\ \vdots \\ p^{n-1}_{0,J-1} \\ p^{n-1}_{1,0} \\ p^{n-1}_{1,1} \\ p^{n-1}_{1,2} \\ \vdots \\ p^{n-1}_{1,J-1} \\ \vdots \\ p^{n-1}_{I-1,0} \\ p^{n-1}_{I-1,1} \\ \vdots \\ p^{n-1}_{I-1,J-1} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c} p \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow 1^{st} \text{ row} \\ \left.\vphantom{\begin{array}{c} p \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow 2^{nd} \text{ row} \\ \left.\vphantom{\begin{array}{c} \vdots \\ p \\ p \\ \vdots \\ p \end{array}}\right\updownarrow I^{th} \text{ row} \end{array} \tag{2.24}$$

By 2.23 we have $\mathbf{p} \in \mathbb{R}^{2IJ\times 1}$. This allows for a linear transformation $\mathbf{p} = \mathbf{Dp}$ that conforms to legal matrix multiplication. Algorithm 1 can be reused by replacing line 4 with $\mathbf{D}$ defined in Algorithm 3 and line 6 with the new $\mathbf{p}$ derived above using Eqn. 2.24. Algorithm 3 is implemented on a supercomputer with an initial interpolated shock pressure wave from [2], and the results are shown in Chapter V.

In the next chapter we will see how the authors of [2] simulate bubble wall velocity.

## 3. Chapter III, Bubble Wall Expansion

Eqn. 12 in [2] states the following:

$$\Delta V_n(t) = (4\pi/3)[R_n^3(t) - R_{na}^3] = kE_L(t) \tag{3.1}$$

where $\Delta V_n(t)$ represents the change in a bubble's volume, $R_n$ is a bubble's radius at the $n^{th}$ time step, $R_{na}$ represents the bubble's initial radius, $k$ is a proportionality constant, and $E_L$ is the energy in a laser pulse. The authors of [2] conduct several experiments with different laser energies and pulse duration values. Table 1 shows the values of $k$ computed given energy $E_L$ and pulse duration $\tau$ using Eqn. 14 in [2].

| $\tau$ (s) | $E_L$ (J) | k |
|---|---|---|
| $6 \times 10^{-9}$ | $1 \times 10^{-3}$ | $6.9377 \times 10^{-8}$ |
| $30 \times 10^{-12}$ | $50 \times 10^{-6}$ | $1.2366 \times 10^{-8}$ |
| $6 \times 10^{-9}$ | 0.01 | $4.45261 \times 10^{-8}$ |
| $30 \times 10^{-12}$ | $1 \times 10^{-3}$ | $1.90789 \times 10^{-8}$ |

**Table 1.** Solving for $k$ in Eqn. 14 of [2] using $\tau$ and $E$. A 3-dimensional least squares analysis will be done to yield an estimator $\hat{k}$

In Figure 2 below, least squares plane is computed and plotted. The equation of the plane defines an estimator for the proportionality constant $k$, $\hat{k}$. This estimator is defined in line 2 of Algorithm 4, and later used on line 4 to determine the final bubble radius value $R_{nb}$. Observe we make an assumption the spot size is equal to the bubble's initial radius $R_{na}$. With $R_{na}$ and $R_{nb}$ known, this is a significant improvement upon the results of [2] because it allows for passing in as input a range of laser energy values $E_L$, laser duration $\tau$, and laser spot size $d$. We are no longer dependent on the proportionality constant $k$, as it can be estimated using the equation of the plane defined in Figure 2.

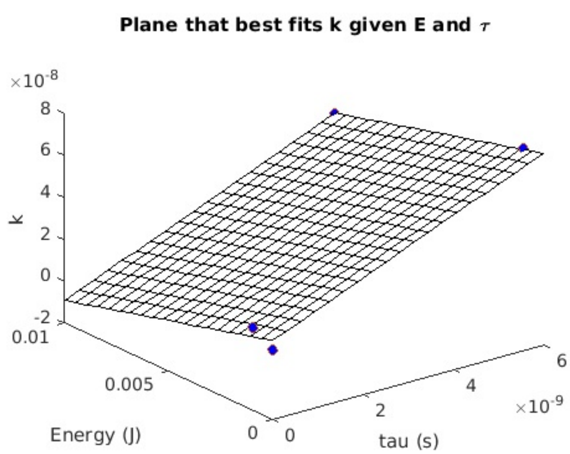



**Figure 2.** The equation of the plane that best fits $k$ is $\hat{k}(E, \tau) = 9.11683\tau - 2.653 \times 10^{-6}E + 16.8421 \times 10^{-9}$. This equation is used in the Bubble Wall Velocity simulation given a laser's spot size, energy, and pulse duration.

**Algorithm 4** Bubble Wall Velocity Simulation given a laser's spot size, energy, and pulse duration

1: **procedure** SIMULATEBUBBLEWITHSPOTSIZEANDENERGY($\tau$, $d$, $E$)

2: $\hat{k} = 9.1168\tau - 2.6530 \times 10^{-6}E + 1.6842 \times 10^{-8}$;

3: $R_{na} = d = R_0 = R_{current}$ ▷ bubble radius initial boundary condition set

4: $R_{nb} = (\frac{3E\hat{k}}{4\pi} + R_{na}^3)^{\frac{1}{3}}$ ▷ Eqn. (14) from [2]

5: $\kappa = \frac{4}{3}$

6: $\rho_0 == 998\frac{kg}{m^3}$

7: $\sigma = 0.072583\frac{N}{m}$

8: $\mu = 0.001046\frac{Ns}{m^3}$

9: $c_0 = 1483\frac{m}{s}$

10: $P_\infty = 100 * 10^3 Pa$

11: $U_0 = 0\frac{m}{s} = U$ ▷ bubble wall velocity initial boundary condition set

12: $n = 7$

13: $B = 314 * 10^6 Pa$

14: <u>Simulation Constant</u>

15: $\Delta t = 1 * 10^{-9}s$

16: <u>Laser Properties</u>

17: **for** t $\leftarrow$ 1 **to** 100ns **do**

18: **if** t $\leq 2\tau$ **then** ▷ if elapsed does time not exceed pulse duration

19: $R_n(t) = \left\{ R_{na}^3 + \frac{R_{nb}^3 - R_{na}^3}{2\tau}\left[t - \frac{\tau}{\pi}\sin\left(\frac{\pi}{\tau}t\right)\right]\right\}^{\frac{1}{3}}$ ▷ Eqn. 17 in [2]

20: **else** $R_n(t) = R_{nb}$ ▷ prevent bubble from growing any further

21: **end if**

22: $R_{current} = R_{current} + \Delta t U$

23: $P = \left(p_\infty + \frac{2\sigma}{R_n}\right)\left(\frac{R_n}{R}\right)^{3\kappa} - \frac{2\sigma}{R} - \frac{4\mu}{R}U$ ▷ Eqn. 7 in [2]

24: $H = \frac{n(p_\infty + B)}{(n-1)\rho_0}\left[\left(\frac{P+B}{p_\infty + B}\right)^{\frac{(n-1)}{n}} - 1\right]$ ▷ Eqn. 10 in [2]

25: $\frac{dP}{dR} = \left(P_\infty + \frac{2\sigma}{R_n}\right)\left((R_n)^{3\kappa}(-3\kappa)R^{-3\kappa-1}\right) + \frac{2\sigma}{R^2} + \frac{4\mu}{R^2}U$ ▷ Deriv. of Eqn. 7 in [2]

26: $\frac{dH}{dR} = \frac{n(P_\infty + B)}{(n-1)\rho_0}\left[\frac{(n-1)}{n}\left(\frac{P+B}{P_\infty + B}\right)^{\frac{n-1}{n}-1} \cdot \left[\frac{1}{P_\infty + B}\frac{dP}{dR}\right]\right]$ ▷ Deriv. of Eqn. 10 in [2]

27: $\dot{U} = \left[-\frac{3}{2}\left(1 - \frac{U}{3C}\right)U^2 + \left(1 + \frac{U}{C}\right)H + \frac{U}{C}\left(1 - \frac{U}{C}\right)R\frac{dH}{dR}\right] \cdot \left[R\left(1 - \frac{U}{C}\right)\right]^{-1}$ ▷ Eqn.5 [2]

28: **end for**

29: **end procedure**

Main result 1 described above is Algorithm 4. Line 2 defines the estimator for $k$ that is derived using a least squares best fit in 3-space. Lines 2 and 3 use the spot size and $\hat{k}$ to define the initial and final bubble radius values required for the simulation. Lines 5-15 define physical constants which are the same ones used in [2]. Line 17 defines a loop that repeats for 100 nanoseconds just like the authors of [2] do. Line 9 checks if the elapsed time has passed the total pulse duration, and if so, does not allow the bubble's radius to grow any further. This makes sense because when there is no laser energy being transferred into the water, there is no reason for the bubble to grow. Line 22 updates the radius at each time step. Line 23 updates the pressure inside the bubble. Line 24 computes the enthalpy at this time step. Line 25 computes the rate of change of bubble pressure with its radius. Line 26 computes the rate of change of enthalpy of the bubble with its radius. Line 27 finally computes the bubble wall velocity at this time step. This whole process is repeated for the next time step until we have reached 100ns.

**Algorithm 5** Find optimal $R_{na}$ and $R_{nb}$ to minimize error between Figures 5 and 9 in [2]

**procedure** FINDOPTIMALRNARNB(figuredata,$\tau$, $lower$R$_{na}$,upper$R_{na}$,lower$R_{nb}$,upper$R_{nb}$, granularity)<br>
 $minError = \infty$<br>
 $minParams = 0$<br>
 **for** $i = lowerR_{na} \longrightarrow upperR_{na}$ **do**<br>
  **for** $j = lowerR_{nb} \longrightarrow upperR_{nb}$ **do**<br>
   **if** $j \geq i$ **then**<br>
    $U \leftarrow simulateBubbleParamWithInitFinalRadius(i, j, \tau)$ ▷ determine bubble wall velocity<br>
    $myError \leftarrow SSE(figuredata, U)$ ▷ determine error using Figure 5 of [2]<br>
    **if** **then**$myError < minError$<br>
     $minError \leftarrow myError$<br>
     $minParams \leftarrow [i, j]$<br>
    **end if**<br>
   **end if**<br>
   $j \leftarrow j + granularity$<br>
  **end for**<br>
  $i \leftarrow i + granularity$<br>
 **end for**<br>
 **return** $minParams$ ▷ $R_{na}$ and $R_{nb}$ with smallest error<br>
**end procedure**

Figures 3-6 together is Main result 2. These figures show that running Algorithm 5 to find optimal initial and final bubble radius values ($R_{na}$ and $R_{nb}$), one can get closer to the experimental data provided in Figure 5 of [2]. Note that the experimental data (colored in red) in Figures 3-6 was extracted using [6], and Algorithm 6 fills in any gaps in data using a simple interpolation technique. Once the experimental data is properly interpolated, it computes a final sum of squared error. In Figure 3, the authors of [2] have an SSE of about 338517, whereas the using the optimal radius values produces an error of about 20655. Dividing these two and then taking the square root gives about 4.04. Therefore using the optimal initial and final radius values produces bubble velocity simulation results that are 4 times closer to the experimental data in Figure 5 than what is reported by the authors of [2]. This kind of significant improvement occurs in the rest of the Figures 4,5,and 6 with orders magnitude 6.2,5.1, and 4.9 respectively.

**Algorithm 6** Find the sum of squared errors between Figure 5 in [2] and simulated bubble wall velocity

**procedure** SSE(figuredata,U)
    $rowSize \leftarrow size(figuredata)$
    $total \leftarrow 0$
    **for** $j = 1 \longrightarrow rowSize$ **do**
        $currentTime \leftarrow 10 \times data(j, 1)$
        $currNS \leftarrow floor(\text{currentTime})$
        $percentagePastNS \leftarrow currentTime - currNS$
        $value = percentagePastIntNS \times U(currNS+1) + (1 - percentagePastIntNS) \times U(currNS)$
        $total = total + (abs(value - data(j, 2)))^2$
    **end for**
**end procedure**

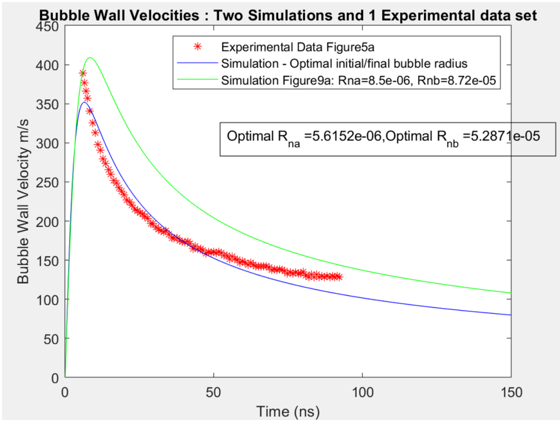



**Figure 3.** The blue curve is closer to the experimental data after 10ns, whereas the green curve is much closer to experimental data just before 10ns. Overall, the blue curve is about 4 times closer to experimental data than the green curve which reproduces results in [2]

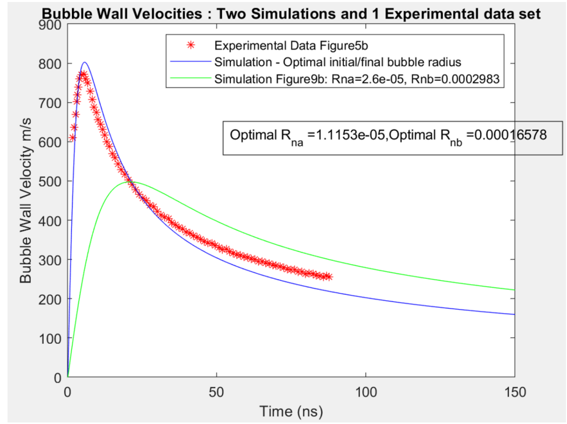



**Figure 4.** Overall, the blue curve is about 6.2 times closer to experimental data than the green curve which reproduces results in [2]

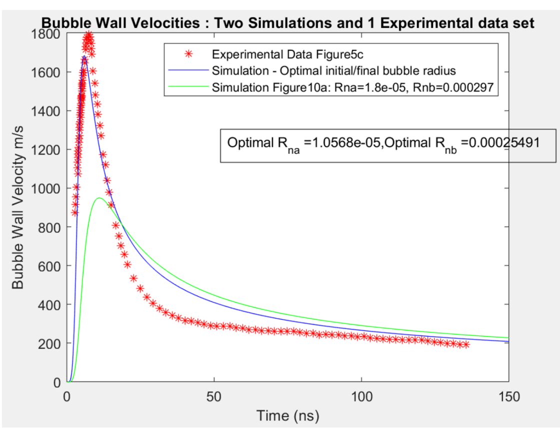



**Figure 5.** The green curve does quite poorly before 20ns, and improves thereafter. However, the blue curve consistently outperforms the green curve overall. The blue curve is closer to the experimental data after 10ns, whereas the green curve is much closer to experimental data just before 10ns. Overall, the blue curve is about 5.1 times closer to experimental data than the green curve which reproduces results in [2]

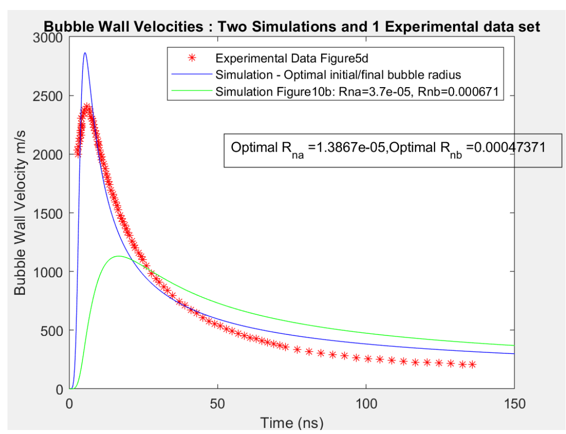



**Figure 6.** The blue curve overestimates the experimental data in the beginning, whereas the green curve under-estimates it. However, after about 20ns, the blue curve gets much closer to the experimental data overall.Overall, the blue curve is about 4.9 times closer to experimental data than the green curve which reproduces results in [2]

# 4. Chapter IV, Bubble Wall Collapse

The derivation in this chapter isolates a bubble's temperature ($T_b$) and pressure ($P_b$) by decoupling the partial differential equations governing bubble collapse as defined in [7]. The main idea of this chapter is to use the same assumptions as Vogel and Parlitz (bubble expansion) in [2], and apply them to Fuster (bubble collapse) in [7]. Together they can describe the entire expand-collapse life cycle of a spherical bubble. Here we assume room temperature is about 300K. In [7], it is assumed the initial pressure is atmospheric pressure. The initial value of for bubble gas temperature $T_b$ and bubble gas density $\rho_b$ are known. The initial pressure for the bubble is 1126 MPa using a 50 $\mu J$ laser pulse energy and 1741 MPa using a 1 mJ laser pulse energy (numbers from [2]). It is also assumed the initial gas velocity $u_b(r, t = 0)$ is known. It is important to remind the reader we are using Eqns. 1-6 in [7] which are better known as equations for continuity, momentum and energy for spherically symmetric motions. The subscript $b$ denotes bubble properties at a particular point in space and time. The bubbles are assumed spherical, which implies no angular dependence. All variables in this derivation depend on the distance from the center of the bubble For example, $u_b(r, t)$ is the radial velocity of gas inside the bubble a distance $r$ away from its center at time $t$.

These symbols are defined in Appendix A.
Define $\Delta t, \Delta r$

$$P_b = \rho_b \frac{R^0}{\overline{\overline{W}}} T_b \tag{4.1}$$

$$P_b = \rho_b \frac{R^0}{\overline{\overline{W}}} T_b \Rightarrow \rho_b = \frac{\overline{W} P_b}{R^0 T_b} \tag{4.2}$$

$$\frac{D\rho_b}{Dt} + \frac{\rho_b}{r^2}\frac{\partial r^2 u_b}{\partial r} = 0 \Rightarrow \left[\frac{D\frac{\overline{W}P_b}{R^0 T_b}}{Dt}\right] + \frac{\frac{\overline{W}P_b}{R^0 T_b}}{r^2}\frac{\partial}{\partial r}(r^2 u_b) = 0 \tag{4.3}$$

Next, evaluate the total derivative.

Recall $\frac{Dx}{Dt} = \frac{\partial x}{\partial t} + (\vec{u_b} \cdot \bigtriangledown)x$ for some arbitrary $x$.

$$\Rightarrow \left[\frac{\partial}{\partial t}(\frac{\overline{W}P_b}{R^0 T_b}) + u_b \cdot \frac{\partial}{\partial r}(\frac{\overline{W}P_b}{R^0 T_b})\right] + \frac{\frac{\overline{W}P_b}{R^0 T_b}}{r^2}\frac{\partial}{\partial r}(r^2 u_b) = 0 \tag{4.4}$$

Next use quotient rule to expand $\frac{\overline{W}P_b}{R^0 T_b}$.

Expand $\frac{\partial}{\partial r}(r^2 u_b)$ using product rule.

$$\Rightarrow \frac{\overline{W}}{R^0}\left[\frac{\frac{\partial P_b}{\partial t}T_b - \frac{\partial T_b}{\partial t}P_b}{(T_b)^2}\right] + u_b\frac{\overline{W}}{R^0}\left[\frac{\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b}{(T_b)^2}\right] + \frac{\frac{\overline{W}}{R^0}\frac{P_b}{T_b}}{r^2}(2ru_b + \frac{\partial u_b}{\partial r}r^2) = 0 \quad (4.5)$$

Note $\frac{\overline{W}}{R^0}$ terms can be canceled above.

Next, multiply across by $(T_b)^2$.

$$\Rightarrow \frac{\partial P_b}{\partial t}T_b - \frac{\partial T_b}{\partial t}P_b = -u_b\left[\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b\right] - \frac{P_b T_b}{r^2}\left[2ru_b + \frac{\partial u_b}{\partial r}r^2\right] \quad (4.6)$$

$$P_b = \rho_b\frac{R^0}{\overline{\overline{W}}}T_b \quad (4.7)$$

$$P_b = \rho_b\frac{R^0}{\overline{\overline{W}}}T_b \Rightarrow \rho_b = \frac{\overline{W}P_b}{R^0 T_b} \quad (4.8)$$

Plug in bubble density $\rho_b$ in Eqn. 4.8 into Eqn. 4.9 (which comes from Eqn. 6 in reference [7] to form Eqn. 4.10.

$$\rho_b\overline{c}_{p,b}\frac{DT_b}{Dt} = \frac{Dp_b}{Dt} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 k_b\frac{\partial T_b}{\partial r}\right) + \frac{4}{3}\mu_b\left(\frac{\partial \mu_b}{\partial r} - \frac{u_b}{r}\right)^2, \quad (4.9)$$

$$\Rightarrow \frac{\overline{W}P_b}{R^0 T_b}\overline{c}_{p,b}\frac{DT_b}{Dt} - \frac{DP_b}{Dt} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 k_b\frac{\partial T_b}{\partial r}\right) + \frac{4}{3}\mu_b\left(\frac{\partial \mu_b}{\partial r} - \frac{\mu_b}{r}\right)^2 \quad (4.10)$$

Next, evaluate the total derivative.

Recall $\frac{Dx}{Dt} = \frac{\partial x}{\partial t} + (\vec{u}_b \cdot \bigtriangledown)x$ for some arbitrary $x$.

$$\Rightarrow \frac{\overline{W}P_b}{R^0 T_b}\overline{c}_{p,b}\left[\frac{\partial T_b}{\partial t} + u_b\cdot\frac{\partial}{\partial r}T_b\right] - \left[\frac{\partial P_b}{\partial t} + u_b\cdot\frac{\partial}{\partial r}P_b\right] = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 k_b\frac{\partial T_b}{\partial r}\right) + \frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r} - \frac{u_b}{r}\right)^2 \quad (4.11)$$

Next, distribute the $\frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}$ term.

$$\Rightarrow \frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}\frac{\partial T_b}{\partial t}+\bar{c}_{p,b}\frac{\overline{W}P_b}{R^0T_b}u_b\cdot\frac{\partial}{\partial r}T_b-\left[\frac{\partial P_b}{\partial t}+u_b\cdot\frac{\partial}{\partial r}P_b\right]=\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2 \tag{4.12}$$

Next, put $\frac{\partial}{\partial r}T_b$ and $\frac{\partial}{\partial r}P_b$ terms on right hand side.

$$\Rightarrow \frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}\frac{\partial T_b}{\partial t}-\frac{\partial}{\partial t}P_b=\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2-\frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_b \tag{4.13}$$

We now have the following two equations:

$$\Rightarrow \frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}\frac{\partial T_b}{\partial t}-\frac{\partial}{\partial t}P_b=\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2-\frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_b \tag{4.14}$$

$$\Rightarrow \frac{\partial P_b}{\partial t}T_b-\frac{\partial T_b}{\partial t}P_b=-u_b\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]-\frac{P_bT_b}{r^2}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right] \tag{4.15}$$

The goal is to solve the two unknown terms $\frac{\partial P_b}{\partial t}$ and $\frac{\partial T_b}{\partial t}$.

First, multiply Equation 4.14 by $T_b$ and add the product to Equation 4.15. This will isolate $\frac{\partial T_b}{\partial t}$.

$$\begin{aligned}\Rightarrow \frac{\overline{W}P_b\bar{c}_{p,b}}{R^0}\frac{\partial T_b}{\partial t}-\frac{\partial P_b}{\partial t}T_b&=\frac{T_b}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_bT_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2\\&\quad-\bar{c}_{p,b}\frac{\overline{W}P_b}{R^0}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_bT_b\\+\left(-\frac{\partial T_b}{\partial t}P_b+\frac{\partial P_b}{\partial t}T_b\right)&=-u_b\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]-\frac{P_bT_b}{r^2}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right]\end{aligned} \tag{4.16}$$

$$
\begin{aligned}
\Rightarrow \frac{\partial T_b}{\partial t}\left[\frac{\overline{W}P_b\bar{c}_{p,b}}{R^0}-P_b\right] =& \frac{T_b}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_bT_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2\\
&-\bar{c}_{p,b}\frac{\overline{W}P_b}{R^0}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_bT_b\\
&-u_b\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]\\
&-\frac{P_bT_b}{r^2}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right]
\end{aligned}
\tag{4.17}
$$

$$
\begin{aligned}
\Rightarrow \frac{\partial T_b}{\partial t} = \frac{1}{\left[\frac{\overline{W}P_bc_{p,b}}{R^0}-P_b\right]}&\left[\frac{T_b}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_bT_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2\right.\\
&-\bar{c}_{p,b}\frac{\overline{W}P_b}{R^0}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_bT_b\\
&-u_b\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]\\
&\left.-\frac{P_bT_b}{r^2}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right]\right]
\end{aligned}
\tag{4.18}
$$

Apply Finite Difference Method, and assume:

$$
\frac{\partial T_b}{\partial t}\approx\frac{T_b(t+\Delta t)-T_b(t)}{\Delta t}
\tag{4.19}
$$

$$
\begin{aligned}
\Rightarrow T_b(t+\Delta t)\approx\frac{1}{\left[\frac{\overline{W}P_bc_{p,b}}{R^0}-P_b\right]}&[\frac{T_b}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_bT_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2\\
&-\frac{\overline{W}P_b}{R^0}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_bT_b\\
&-u_b\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]\\
&-\frac{P_bT_b}{r^2}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right]]\Delta t+T_b(t)
\end{aligned}
\tag{4.20}
$$

We now have the following two equations:

$$
\Rightarrow\frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}\frac{\partial T_b}{\partial t}-\frac{\partial}{\partial t}P_b=\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2-\frac{\overline{W}P_b}{R^0T_b}\bar{c}_{p,b}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_b
\tag{4.21}
$$

$$\Rightarrow \frac{\partial P_b}{\partial t}T_b - \frac{\partial T_b}{\partial t}P_b = -u_b\left[\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b\right] - \frac{P_bT_b}{r^2}\left[2ru_b + \frac{\partial u_b}{\partial r}r^2\right] \tag{4.22}$$

The goal is to solve the two unknown terms $\frac{\partial P_b}{\partial t}$ and $\frac{\partial T_b}{\partial t}$.

Next, multiply Equation 4.22 by $\frac{\overline{W}\bar{c}_{p,b}}{R^0T_b}$ and add the product to Equation 4.21.
This will isolate $\frac{\partial P_b}{\partial t}$.

$$\begin{aligned}\Rightarrow \frac{\partial P_b}{\partial t}T_b\frac{\overline{W}\bar{c}_{p,b}}{R^0T_b} - \frac{\partial P_b}{\partial t} =& - u_b\frac{\overline{W}\bar{c}_{p,b}}{R^0T_b}\left[\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b\right] - \frac{P_bT_b\overline{W}}{r^2}\frac{\bar{c}_{p,b}}{R^0T_b}\left[2ru_b + \frac{\partial u_b}{\partial r}R^2\right]\\ &+ \left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)\right.\\ &+ \left.\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r} - \frac{u_b}{r}\right)^2 - \bar{c}_{p,b}\frac{\overline{W}P_b}{R^0T_b}u_b\cdot\frac{\partial}{\partial r}T_b + u_b\cdot\frac{\partial}{\partial r}P_b\right)\end{aligned} \tag{4.23}$$

$$\begin{aligned}\Rightarrow \frac{\partial P_b}{\partial t}\left[\frac{\overline{W}c_{p,b}}{R^0} - 1\right] =& - u_b\frac{\overline{W}\bar{c}_{p,b}}{R^0T_b}\left[\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b\right] - \frac{P_bT_b\overline{W}}{r^2}\frac{\bar{c}_{p,b}}{R^0T_b}\left[2ru_b + \frac{\partial u_b}{\partial r}r^2\right]\\ &+ \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right) + \frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r} - \frac{u_b}{r}\right)^2 - \bar{c}_{p,b}\frac{\overline{W}P_b}{R^0T_b}u_b\cdot\frac{\partial}{\partial r}T_b\\ &+ u_b\cdot\frac{\partial}{\partial r}P_b\end{aligned} \tag{4.24}$$

$$\begin{aligned}\Rightarrow \frac{\partial P_b}{\partial t} =& \frac{1}{\frac{\overline{W}\bar{c}_{p,b}}{R^0} - 1}\left[-u_b\frac{\overline{W}\bar{c}_{p,b}}{R^0T_b}\left[\frac{\partial P_b}{\partial r}T_b - \frac{\partial T_b}{\partial r}P_b\right] - \frac{P_bT_b\overline{W}}{r^2}\frac{\bar{c}_{p,b}}{R^0T_b}\left[2ru_b + \frac{\partial u_b}{\partial r}r^2\right]\right.\\ &\left.+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right) + \frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r} - \frac{u_b}{r}\right)^2 - \bar{c}_{p,b}\frac{\overline{W}P_b}{R^0T_b}u_b\cdot\frac{\partial}{\partial r}T_b + u_b\cdot\frac{\partial}{\partial r}P_b\right]\end{aligned} \tag{4.25}$$

Just as before, use Finite Difference Method for approximating $\frac{\partial P_b}{\partial t}$

$$\frac{\partial P_b}{\partial t} \approx \frac{P_b(t + \Delta t) - P_b(t)}{\Delta t}$$

$$\Rightarrow P_b(t+\Delta t) \approx \frac{1}{\frac{\overline{W}P_b\bar{c}_{p,b}}{R^0}-1}\left[-u_b\frac{\overline{W}P_b\bar{c}_{p,b}}{R^0T_b}\left[\frac{\partial P_b}{\partial r}T_b-\frac{\partial T_b}{\partial r}P_b\right]-\frac{P_bT_b\overline{W}}{r^2}\frac{P_b\bar{c}_{p,b}}{R^0T_b}\left[2ru_b+\frac{\partial u_b}{\partial r}r^2\right]\right.$$
$$\left.+\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2k_b\frac{\partial T_b}{\partial r}\right)+\frac{4}{3}\mu_b\left(\frac{\partial u_b}{\partial r}-\frac{u_b}{r}\right)^2-\frac{\overline{W}P_b}{R^0T_b}u_b\cdot\frac{\partial}{\partial r}T_b+u_b\cdot\frac{\partial}{\partial r}P_b\right]\Delta t$$
$$+P_b(t) \tag{4.26}$$

$$P_b=\rho_b\frac{R^0}{\overline{W}}T_b \tag{4.27}$$

$$P_b=\rho_b\frac{R^0}{\overline{W}}T_b \Rightarrow \rho_b=\frac{\overline{W}P_b}{R^0T_b} \tag{4.28}$$

Plug in bubble density $\rho_b$ in Eqn. 4.28 into Eqn. 5 from reference [7] to form Eqn. 4.29.

$$\begin{aligned}&\rho_b\frac{Du_b}{Dt}=-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_bu_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial\mu_b}{\partial r}\\ &\Rightarrow\left[\frac{\overline{W}P_b}{R^0T_b}\right]\frac{Du_b}{Dt}=-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_bu_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial\mu_b}{\partial r}\end{aligned} \tag{4.29}$$

Next, evaluate the total derivative.

Recall $\frac{Dx}{Dt}=\frac{\partial x}{\partial t}+(\vec{u}_b\cdot\bigtriangledown)x$ for some arbitrary $x$.

$$\Rightarrow\frac{\overline{W}P_b}{R^0T_b}\left[\frac{\partial u_b}{\partial t}+u_b\cdot\frac{\partial}{\partial r}u_b\right]=-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_bu_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial\mu_b}{\partial r} \tag{4.30}$$

Assume $\frac{\partial u_b}{\partial t}\approx\frac{u_b(t+\Delta t)-u_b(t)}{\Delta t}$

$$\Rightarrow\frac{\overline{W}P_b}{R^0T_b}\left[\frac{u_b(t+\Delta t)-u_b(t)}{\Delta t}+u_b\cdot\frac{\partial}{\partial r}u_b\right]\approx-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_bu_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial\mu_b}{\partial r} \tag{4.31}$$

$$\Rightarrow \frac{u_b(t+\Delta t)-u_b(t)}{\Delta t}+u_b\cdot\frac{\partial}{\partial r}u_b \approx \frac{R^0 T_b}{\overline{W} P_b}\left(-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_b u_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial \mu_b}{\partial r}\right) \tag{4.32}$$

$$\Rightarrow u_b(t+\Delta t) \approx \left[\frac{R^0 T_b}{\overline{W} P_b}\left(-\frac{\partial p_b}{\partial r}+\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)-\frac{8}{3}\frac{\mu_b u_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial \mu_b}{\partial r}\right)-u_b\cdot\frac{\partial}{\partial r}u_b\right]\Delta t+u_b(t) \tag{4.33}$$

But recall that $-\frac{\partial P_b}{\partial r} \approx -\frac{(P_b(r+\Delta r,t)-P_b(r,t))}{\Delta r}$

By the product rule:

$$\frac{4}{3r^2}\frac{\partial}{\partial r}\left(r^2\mu_b\frac{\partial u_b}{\partial r}\right)=\frac{4}{3r^2}\left[2r\mu_b\frac{\partial u_b}{\partial r}+r^2\mu_b\frac{\partial^2 u_b}{\partial r^2}+r^2\frac{\partial \mu_b}{\partial r}\frac{\partial u_b}{\partial r}\right] \tag{4.34}$$

Now we can rewrite Eqn. 4.33 as:

$$\begin{aligned}u_b(t+\Delta t) \approx \Bigg[\frac{R^0 T_b}{\overline{W} P_b}\Bigg(&-\frac{(P_b(r+\Delta r,t)-P_b(r,t))}{\Delta r}+\frac{4}{3r^2}\left[2r\mu_b\frac{\partial u_b}{\partial r}+r^2\mu_b\frac{\partial^2 u_b}{\partial r^2}+r^2\frac{\partial \mu_b}{\partial r}\frac{\partial u_b}{\partial r}\right]\\ &-\frac{8}{3}\frac{\mu_b u_b}{r^2}-\frac{4}{3}\frac{u_b}{r}\frac{\partial \mu_b}{\partial r}\Bigg)-u_b\cdot\frac{\partial}{\partial r}u_b\Bigg]\Delta t+u_b(t)\end{aligned} \tag{4.35}$$

## 5. Chapter V, Conclusion

Chapter II describes how the viscous wave equation can be solved and implemented in 1 and 2 dimensions. Chapters III and IV discuss the expansion-collapse life cycle of a pressure bubble generated by a laser pulse. Eqn. 1 in [2], however, gives a formula for shock wave velocity. This velocity is super sonic and would reach a hydrophone more quickly than the bubble wall velocities described in Chapters III and IV. Therefore, the shock wave velocity is chosen to perform numerical integration in order to determine how far the shock pressure wave travels over a period of time. Specifically, performing numerical integration on Eqn. 1 of [2] over $[0, i\Delta t]$ for $1 \leq i \leq 150$ gives $r(t_i)$, the distance traveled by the pressure wave at the $i^{th}$ time step. By filling in gaps between time steps using interpolation, we get a smooth initial shape of the shock pressure wave. This initial pressure wave is used to initialize a simulation of the 2-dimensional viscous wave equation from Chapter II to form the image shown in Figure 7. After 43 nanoseconds, the pressure waves looks like Figure 8. The value of running a simulation like this is to get a better idea of how far a shock pressure wave can travel in a viscous medium like water. We can run simulations with varying energy, spot size, and pulse duration and get a good idea of how far a pressure wave can propagate before procuring expensive laser equipment. Furthermore, we can extend this work to include communication protocols such as pulse position modulation or pulse amplitude modulation. Lastly, we can add noise to the pressure wave to simulate real-life scenarios. These noisy pressure waves can be simulated over a period of time and detected by signal processing software to decode into 0s and 1s. Decoding into 0s and 1s can lead to an efficient communication protocol across the air-water boundary.

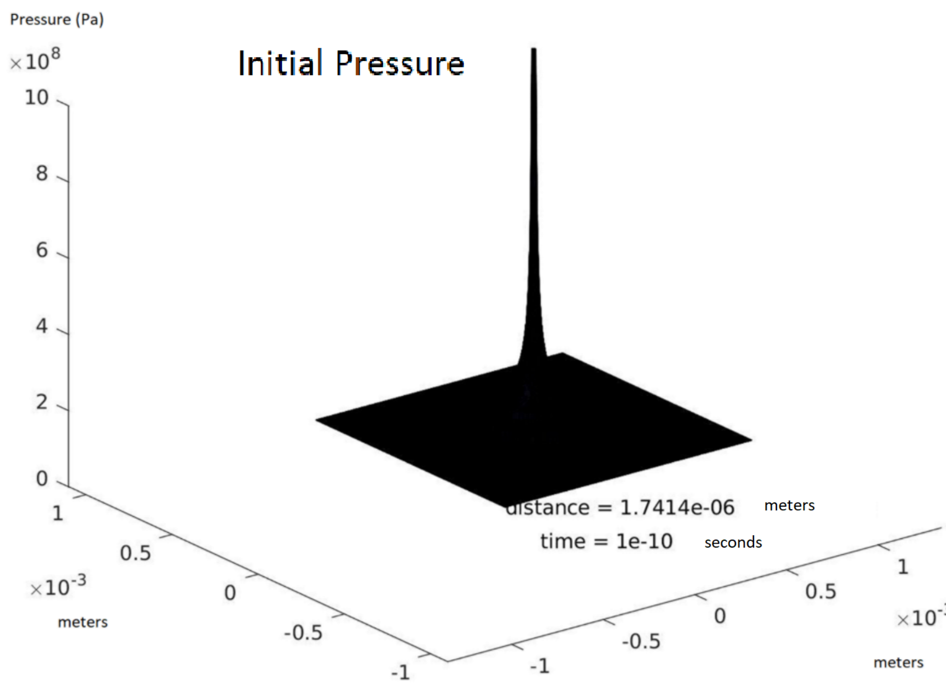
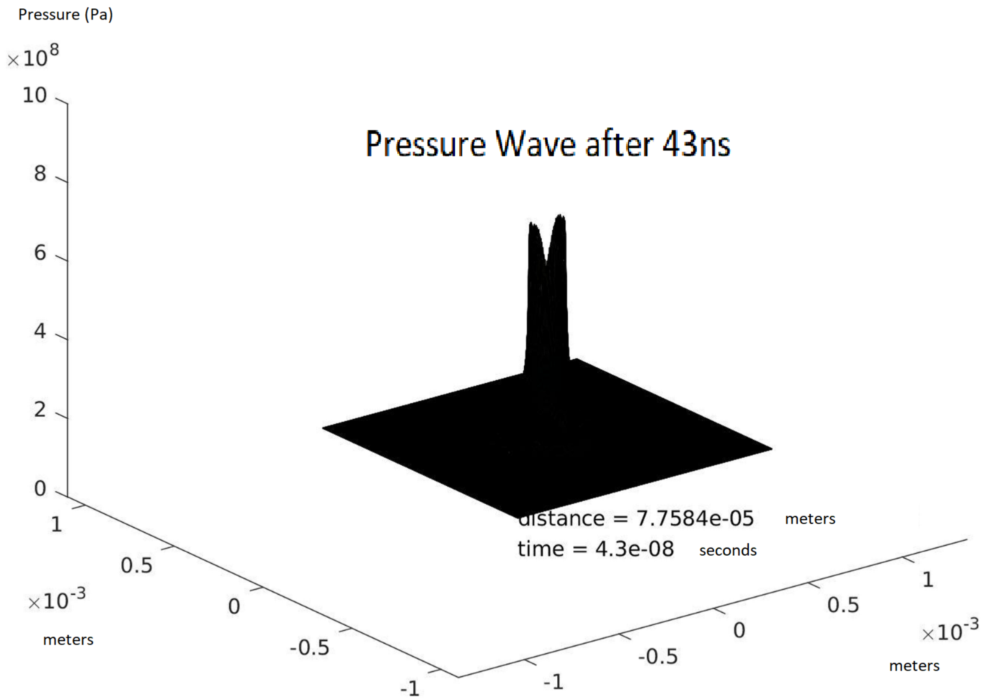



**Figure 7.** The shape of the wave shown here at the initial time step was determined through numerical integration of Eqn. 1 of [2], the shock wave velocity. Since the shock wave velocity is super sonic and bubble wall velocities are not, a shock pressure wave would reach a hydrophone more quickly than subsequent bubble wall collapse pressure waves. This is the reason why the shock velocity was chosen over bubble wall velocities for numerical integration. In the future, we can extend this simulation to include numerical integration of bubble wall velocity using the same interpolation technique to produce a more realistic simulation.




## References


[1] A.G. Bell,"On the production and reproduction of sound by light" Am. J. Sci. Vol. 20, 305-324 (1880)

[2] A.Vogel, S. Busch, U. Parlitz, "Shock wave emission and cavitation bubble generation by picosecond and nanosecond optical breakdown in water" The Journal of the Acoustical Society of America 100, 148 (1996)

[3] T. Jones, D. Kaganovich, M. Helle, T. Ting, B. Hafizi, J. Peñano, "Underwater Laser Acoustic Source Control Division Using Shaped Plasmas" 169th Meeting of the Acoustical Society of America (2015)

[4] Shu-Jen Steven Tsai, Power Transformer Partial Discharge (PD) Acoustic Signal Detection using Fiber Sensors and Wavelet Analysis, Modeling, and Simulation, Chapter 2, M.S. Thesis, Virginia Polytechnic Institute and State University, 2002. Online at https://vtechworks.lib.vt.edu/handle/10919/35983.

[5] D.M. Causon, C.G. Mingham. "Introductory Finite Difference Methods for PDE" Textbook appendix D.5 Appendix D.5 Theory for Dominant Eigenvalues pg 139.

[6] Ankit Rohatgi, "WebPlotDigitizer" Website: https://apps.automeris.io/wpd

[7] D. Fuster, C. Dopazo, G. Hauke, "Liquid compressibility effects during the collapse of a single cavitating bubble" Acoustical Society of America (2011)

[8] L. Kinsler, A. R. Frey, A. Coppens, J. Sanders, "Fundamentals of Acoustics 4th Editions" Textbook (2000)

[9] M. Islam, M. Younis, "An Adaptive DPPM for Efficient and Robust Visible Light Communication Across the Air-Water Interface" UMBC Department of Electrical Engineering

## Appendix A.

| | |
|---|---|
| $\overline{W}$ | Averaged molecular weight of gas |
| $\rho_b$ | Bubble density |
| $P_b$ | Pressure inside bubble |
| $R^0$ | Universal perfect gas constant |
| $T_b$ | Bubble temperature |
| $r$ | Distance from center of bubble |
| $t$ | Time |
| $u_b(r,t)$ | Radial velocity of gas inside the bubble a distance $r$ away from its center at time $t$ |
| $\mu_b$ | Viscosity coefficient |
| $\tau_{p,b}$ | Specific heat at constant pressure of the bubble |
| $k_b$ | Gas thermal conductivity |