

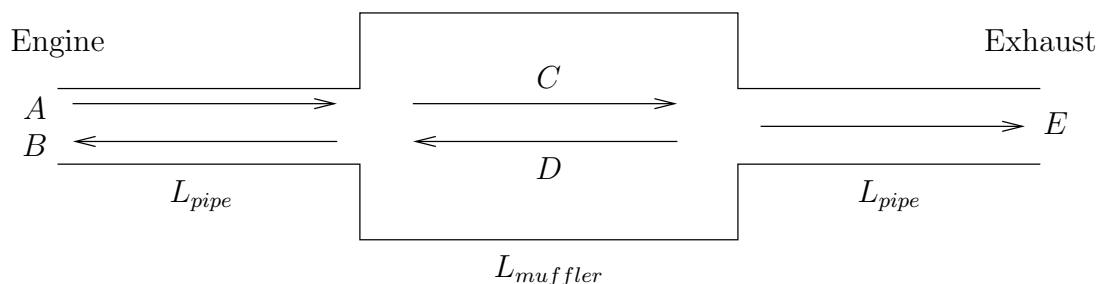
Scientific Programming

Project 4: Muffler Simulation

Due 11/11. You can work on this in teams of two.

Project Goal To simulate sound waves going through a car muffler and determining how much the sound decreases at different frequencies.

Background A basic reactive muffler is a hollow chamber that is larger than the exhaust tube that reflects sound waves back and forth. At some frequencies the sound wave bouncing back and forth cancel each other out and reflect the sound back towards the car engine allowing only some of the noise to come out of the exhaust pipe.



In the above figure of the muffler there are five different sound waves we will consider.

A	Source noise from the car engine
B	Sound reflected back towards the car engine
C	Sound traveling forward along the muffler
D	Sound traveling backward along the muffler
E	Sound coming out of the exhaust pipe

The exhaust pipes and muffler have circular cross sections with radii R_{pipe} and $R_{muffler}$, respectively. Let S_{pipe} and $S_{muffler}$ be the areas of their cross sections.

Each sound wave satisfies the wave equation, so except at their ends the values of each can be found using the update rules we calculated in class:

$$\begin{aligned}
 A(x, t + \Delta t) &= r^2(A(x + \Delta x, t) + A(x - \Delta x, t)) + 2(1 - r^2)A(x, t) - A(x, t - \Delta t) \\
 B(x, t + \Delta t) &= r^2(B(x + \Delta x, t) + B(x - \Delta x, t)) + 2(1 - r^2)B(x, t) - B(x, t - \Delta t) \\
 C(x, t + \Delta t) &= r^2(C(x + \Delta x, t) + C(x - \Delta x, t)) + 2(1 - r^2)C(x, t) - C(x, t - \Delta t) \\
 D(x, t + \Delta t) &= r^2(D(x + \Delta x, t) + D(x - \Delta x, t)) + 2(1 - r^2)D(x, t) - D(x, t - \Delta t) \\
 E(x, t + \Delta t) &= r^2(E(x + \Delta x, t) + E(x - \Delta x, t)) + 2(1 - r^2)E(x, t) - E(x, t - \Delta t)
 \end{aligned}$$

where $r = \frac{c\Delta t}{\Delta x}$.

At the end of each wave we will use the transparent sound conditions discussed in the handout:

$$\begin{aligned}
 A(L_{pipe}, t + \Delta t) &= \frac{2r^2 A(L - \Delta x, t) + 2(1 - r^2)A(L, t) + (r - 1)A(L, t - \Delta t)}{r + 1} \\
 B(L_{pipe}, t + \Delta t) &= \frac{2r^2 B(L - \Delta x, t) + 2(1 - r^2)B(L, t) + (r - 1)B(L, t - \Delta t)}{r + 1} \\
 C(L_{muffler}, t + \Delta t) &= \frac{2r^2 C(L - \Delta x, t) + 2(1 - r^2)C(L, t) + (r - 1)C(L, t - \Delta t)}{r + 1}
 \end{aligned}$$

$$D(L_{muffler}, t + \Delta t) = \frac{2r^2 D(L - \Delta x, t) + 2(1 - r^2) D(L, t) + (r - 1) D(L, t - \Delta t)}{r + 1}$$

$$E(L_{pipe}, t + \Delta t) = \frac{2r^2 E(L - \Delta x, t) + 2(1 - r^2) E(L, t) + (r - 1) E(L, t - \Delta t)}{r + 1}$$

The engine noise will be modeled as a single frequency tone with frequency ω . This gives us the following formula for the start of sound wave A :

$$A(0, t) = \sin(2\pi\omega t)$$

The starting values of the other four waves are based on how much sound is reflected or transmitted at each end of the muffler. The equations are

$$B(0, t + \Delta t) = \frac{(1 - \alpha)A(L_{pipe}, t + \Delta t) + 2\alpha D(L_{muffler}, t + \Delta t) - A(L_{pipe}, t - \Delta t) + B(0, t - \Delta t) + \alpha C(0, t - \Delta t) - \alpha D(L_{muffler}, t - \Delta t)}{1 + \alpha}$$

$$C(0, t + \Delta t) = \frac{2A(L_{pipe}, t + \Delta t) - (1 - \alpha)D(L_{muffler}, t + \Delta t) - A(L_{pipe}, t - \Delta t) + B(0, t - \Delta t) + \alpha C(0, t - \Delta t) - \alpha D(L_{muffler}, t - \Delta t)}{1 + \alpha}$$

$$D(0, t + \Delta t) = \frac{(1 - \alpha^{-1})C(L_{muffler}, t + \Delta t) - C(L_{muffler}, t - \Delta t) + D(0, t - \Delta t) + \alpha^{-1}E(0, t - \Delta t)}{1 + \alpha^{-1}}$$

$$E(0, t + \Delta t) = \frac{2C(L_{muffler}, t + \Delta t) - C(L_{muffler}, t - \Delta t) + D(0, t - \Delta t) + \alpha^{-1}E(0, t - \Delta t)}{1 + \alpha^{-1}}$$

where $\alpha = \frac{S_{muffler}}{S_{pipe}}$.

For a given frequency, we can measure how much the muffler reduces the sound level (in decibels) for the time interval t_0 to t_1 using the following formula:

$$-10 \log_{10} \frac{\int_{t_0}^{t_1} A(0, t)^2 dt}{\int_{t_0}^{t_1} E(L_{pipe}, t)^2 dt}$$

If you record the input and output signal at each time step and store them in the arrays `inputSignal` and `outputSignal` then the above formula can be estimated as

$$-10 * \log_{10} (\text{sum}(\text{outputSignal}.^2) / \text{sum}(\text{inputSignal}.^2))$$

Project Assignment You will be simulating a muffler with the following parameters:

Length of pipe	L_{pipe}	50 cm
Length of muffler	$L_{muffler}$	70 cm
Radius of pipe	R_{pipe}	2.5 cm
Radius of muffler	$R_{muffler}$	8 cm
Speed of sound	c	340 m/s
Segment length	Δx	.01 m
Time step	Δt	.00001 s

You need to calculate how much the muffler reduces the signal strength for frequencies ranging from 10 Hz to 1000 Hz. You should graph this and answer the following questions:

1. What frequencies did the muffler do the best job at filtering out?
2. What frequencies did the muffler do the worst job at filtering out?
3. Based on the graph that you have made, do you see a pattern that would let you predict what would happen at higher frequencies?

Requirements

- You must write a function `signalLoss` that takes a frequency as an input and outputs how much the sound level is decreased (in dB).
- The function should perform is simulation for a half-second with the sound loss calculated over the last quarter second.
- You should record the resulting sound losses at a minimum of 10 Hz intervals.

Hints and Suggestions

- Look at the sample code for sound traveling from a small tube to a big tube in the handout. This will be very helpful.
- First, write code similar to the one in the handout where you enter the frequency and can watch the graphs to ensure that everything works properly.
- For checking your work: at a frequency of 242 Hz there should be almost not loss in sound volume and at a frequency of 720 Hz there should be a sound loss of approximately 13.2 dB.
- After you have it working for testing a single frequency, rewrite you code to remove the plotting and turn it into a function.
- Use this function within a loop to calculate how much the sound level decreases for different frequencies.
- The total running time for creating the graph could take anywhere from 15 minutes to an hour, so you should run it on campus on a good computer and not over a remote connection.