P441/442: Open Ended Lab, 2021-22

Gravity-driven oscillations of a liquid column in a drinking straw

October 10, 2021

Soumik Bhattacharyya
Roll No.- 1811155
Fourth year Integrated M.Sc

Lab Instructors:
Dr. Pratap Kumar Sahoo
Dr. Gunda Santosh Babu

School of Physical Sciences
National Institute of Science Education and Research
Bhubaneswar

Gravity-driven oscillations of a liquid column in a drinking straw

Soumik Bhattachryya
SPS, NISER

Abstract
When a vertical cylindrical small pipe is submerged into a fluid with its upper end closed, it feels a hydrostatic force from the liquid column inside. Now, if the pressure it released, the fluid column rises in the upward direction and then again comes down because of gravitational force. These two forces maintains a oscillation of the peak of the surface of the fluid inside the pipe, while another force of damping decreases the overall oscillation amplitude and finally stops it. This phenomenon was studied earlier and two different models, Newtonian model and Lorenceau model, have been developed. In this experiment, these two models have been studied and a comparison between the two has been drawn. Data has been taken using two different fluids (water and alcohol solution) and two different pipe materials (plastic and glass). The damping for using plastic was much greater than that of glass. In general, the Lorenceau model has been found to be a better approach to solve this problem. Furthermore, a power spectrum has been produced using Fast Fourier Transformation (FFT) and it has been found out that this system follows Hooke's Law for small oscillations.

Objectives

1. To model gravity-driven fluid oscillation in pipe using two different models: Newtonian Model and Lorenceau Model.
2. To calculate and compare the coefficients of damping in different experiment setup; a) using two different liquids (water and alcohol solution) and b) using two different materials for straw (plastic and glass).
3. To generate power spectrum and calculate the natural frequency.
4. To calculate the acceleration due to gravity (g) from the value of natural frequency and verify Hooke's Law for small oscillations.

Introduction

When a vertical cylindrical both-side opened pipe is closed at its top and then submerged in a liquid, it pushes the liquid surface inside of pipe, following fluid statics. The surface of liquid inside the pipe is now at a lower level than the rest of the liquid surface. When the pressure of the pipe is lifted suddenly, i.e. the closed side of the pipe is opened, the fluid rushes into the pipe in upward direction, goes above the rest of the surface and again comes down due to gravity. This process of the fluid going above the rest of the surface and again coming below it continues infinitely, unless there is a damping force, that acts against this oscillation. In practical examples, this damping force is greater than the driving forces, i.e. gravitational force and the force due to hydrostatic pressure. If we take a combination of fluid and pipe material such that the viscosity and surface tension is very low, then it reduces the damping force and we can observe more number of oscillations before the entire fluid comes to equal height.

Although this problem sounds quite simple, in reality it is much more difficult to model. It is case of damped oscillation, where the mass of the system, i.e. the driving force also varies with time, making it a more complex and unique situation. However, two theoretical models to solve this problem have been developed earlier. The first one was developed in 2001 by *Lorenceau et al.*[1], which we shall be referring to as **Lorenceau model**. A much simpler, yet quite efficient model using Newtonian Mechanics was developed by *Smith et al.*[2] in 2018.

We shall refer to that as **Newtonian Model**.

In this report, the experimental data using two types of straw (plastic and glass) and two types of liquid (water and alcohol solution) has been modelled using the two theoretical models. A comparison between the two has been drawn. Later, the Lorenceau model has been used to plot the power spectrum and to show that this oscillation follows Hooke's Law for small oscillation. It is also verified by calculating the accelaration due to gravity for data from both the liquids.

Theory

Newtonian Model

In the Newtonian model, the forces along axis has been considered to get an equation of motion. The forces working are, gravitational force, hydrostatic pressure and damping force. Note that the damping can rise from either any or all of the internal friction effects of the tube (or straw), surface roughness, development length effects, transition to turbulence or capillary effects; but we are taking only one force term equivalent to all of the reasons along the length of the tube. The direction of gravitational force is downward and hydrostatic pressure works in upward direction.

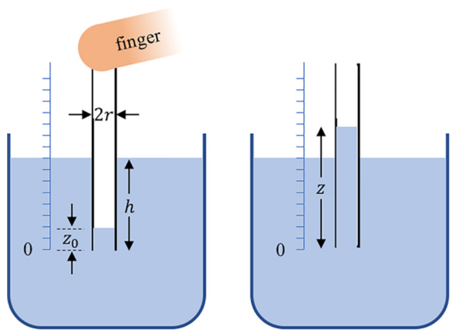


Figure 1: The left figure is when the straw is closed by a finger, the right side figure is when the pressure is lifted. This image is taken from *Smith et al. (2018)* [2]

Now, if z is the height of liquid column at some instance, z_0 is the initial height, H is length of pipe, h is the height of pipe that is dipped inside fluid, ρ is the density of liquid, g is the accelaration due to gravity, A is cross-section area of pipe, m is the mass of liquid inside pipe, b' is damping parameter; then we can get the following relations.

$$z_0 = H \times \frac{1 - P_{atm}}{\rho \times h} \tag{1}$$

We get this relation from fluid statics. Here P_{atm} is the atmospheric pressure. Now the forces are given by (in the upward direction),

$$\text{Force from Hydrostatic Pressure} = \rho g h A$$

$$\text{Gravitational Force} = -mg$$

$$\text{Damping Force} = b'\dot{z}$$

Now, we get the equation of motion by merging them all as the first derivative of momentum, where momentum is $p = m\dot{z} = \rho g A\dot{z}$.

$$\frac{dp}{dt} = \rho g h A - mg + b'\dot{z} \tag{2}$$

$$\Rightarrow \quad m\ddot{z} + \dot{m}\dot{z} = \rho g h A - mg + b'\dot{z} \tag{3}$$

$$\Rightarrow \quad \ddot{z} = -\frac{1}{z}(\dot{z}^2 + gz - gh + b\dot{z}) \tag{4}$$

Note that b' is replaced by bρA to simplify the equation.

Lorenceau model

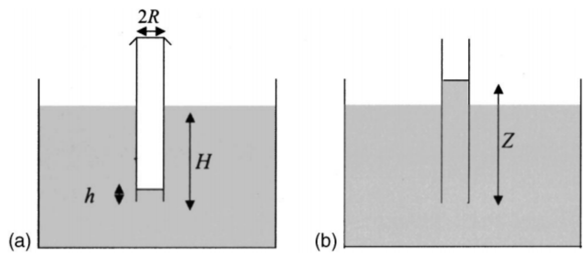


Figure 2: (a) The straw is closed-ended, (b) Oscillation begins when the pressure is lifted. This image is taken from *Lorenceau et al. (2001)* [1]

The Lorenceau model is developed on the basis of energy conservation, not force. We follow the notations used in the paper [1]. The radius of straw is taken R, the height of straw inside water is H, the height of fluid column at an instant is Z.

Now, The radius of the straw is much larger than the capillary length, so the capillary forces can be neglected and the main driving force is the hydrostatic pressure,

$$F = \rho g \pi R^2 (H - Z) \tag{5}$$

We get the potential energy by integrating force equation,

$$U = \frac{1}{2}\rho g \pi R^2 - \rho g H \pi R^2 Z \tag{6}$$

Now, the total energy (E) is found by adding the kinetic energy.

$$E = \frac{1}{2}\rho \pi R^2 Z \dot{Z}^2 + \frac{1}{2}\rho g \pi R^2 - \rho g H \pi R^2 Z \tag{7}$$

we can divide both side by a factor of $\rho \pi R^2 H$ to get rid of the extra terms.

$$e = E/\rho \pi R^2 H = \frac{1}{2} z \dot{z}^2 + \frac{1}{2} z^2 - z \tag{8}$$

Now, we can add the damping terms. The two main reason of damping are namely viscous dissipation and singular pressure loss. The viscous dissipation can be neglected for a short period of time. However we cannot neglect the singular pressure loss. It rises due to difference in radius of the straw and the beaker, which causes eddies at the entrance, thus gives rise to a dissipating energy. It is given by,

$$\Delta P = \frac{1}{2}\rho \dot{Z}^2 \tag{9}$$

Now, the pressure loss is always positive, so the energy of the system always decreases. This leads us to equation of de,

when the liquid is rising (dz > 0),

$$de = \frac{1}{2}\dot{z}^2 dz \tag{10}$$

and when liquid is falling (dz < 0),

$$de = -\frac{1}{2}\dot{z}^2 dz \tag{11}$$

From equation (8),
For dz > 0

$$z\ddot{z} + \dot{z} = 1 - z \tag{12}$$

for dz < 0

$$z\ddot{z} = 1 - z \tag{13}$$

We come to this derivation by neglecting the viscous dissipation for small time scale, but we need to add a term for that for a large time frame. The viscous dissipation is given by the Poiseuille friction (Ω), which is given by,

$$\Omega = \frac{16\eta H^{1/2}}{\rho R^2 g^{1/2}} \tag{14}$$

where η is the viscosity of the liquid. Now, equation (12) and (13) becomes,
For dz > 0

$$z\ddot{z} + \dot{z} = 1 - z - \Omega z\dot{z} \tag{15}$$

for dz < 0

$$z\ddot{z} = 1 - z - \Omega z\dot{z} \tag{16}$$

These are the working equations for Lorenceau Model.

Small Oscillations

An extension of the Newtonian model show us that this experiment follows Hooke's Law for small oscillations. Within the limit of y«h, the equation of motion for Newtonain model (eq. 4) simplifies to,

$$\ddot{y} = -\frac{g}{h} y \tag{17}$$

The natural frequency of this system can be found out by,

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi}\left(\frac{g}{h}\right)^{1/2} \tag{18}$$

We can analytically calculate this and verify the value by plotting a power spectrum.

Power Spectrum

To get the natural frequency of the system from our data, we need to do a Fourier transform from time domain to frequency domain. As we have a set of discrete datapoints, we go for Discrete Fourier Transformation and it is done by doing an even faster method of FFT, by using the `numpy.fft` module of python. The process has been discussed in *Appendix D*.

The data in frequency domain is now fitted with a Lorentzian function to get the peak. A Lorentzian function is defined as the Fourier-transform of an exponentially decaying oscillation, which is why this was used. The general lorentzian function is given by,

$$y = \frac{A}{1 + \left(\frac{x - x_0}{\gamma}\right)^2} \tag{19}$$

where A is the amplitude (height) of the peak, x_0 is the position of the peak and γ is the peak half width at half of maximum level. The value of x_0 is nothing but the desired natural frequency of the system.

Now, we can find the natural frequency for different heights. We can get a relation between natural frequency and height of pipe submerged in fluid.

$$f_0^2 = \frac{g}{4\pi^2}\frac{1}{h} \tag{20}$$

By plotting a curve between f_0^2 and $\frac{1}{h}$, we can get the slope and verify the value of g.

Apparatus used

- Glass pipe and plastic pipe
- Beaker
- Water and Ethyl Alcohol (95%)
- Mobile camera (120 fps)
- Stand for mobile phone
- Water Colour for tracking water level from video

Data Collection

The setup for this experiment is quite simple. The liquid was taken in a beaker and it was colourised as coloured liquid is easier to track in video. Then the pipe was closed from one end just by finger and it was dipped into fluid. The camera was set in high frame rate mode and placed in front of the beaker. After starting the recording, we loose the pressure from the end of the pipe which was outside the liquid. The oscillations begin and we capture it in the video.

For our case, videos were taken in 90 and 120 fps. We used two different pipes, one of glass (height= 61.5 cm, radius= 0.55 cm) and one of plastic (height= 13.95 cm, radius= 0.523 cm). We used the glass pipe for varying the liquid. We did the experiment with water and a solution of 50% water and 50% ethyl alcohol (95%). Data was taken by submerging different heights of pipe inside water. The videos are added in this drive folder, `https://drive.google.com/drive/folders/15ireubu8UAl0MnhBZCov3Ko0br0EvcsK?usp=sharing`

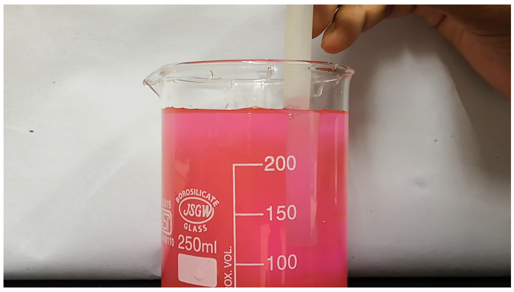


Figure 3: Data Collection

- **Tracking the data**

After that the data was tracked using the free software `tracker`. In this software, we can calibrate the length in general using the diameter of beaker or any known length. Once we start tracking in this software, it pauses the video in every frame and we can select the height of the liquid at that frame. At the end we have two lists of all the time frames and the desired heights. This data can be saved as `.txt` file. The datafiles are also submitted in the above drive folder.

Data Analysis

Calibration

The data collected from tracker is tracked manually. So, it is very possible that some of the time frames before starting of the oscillation are also included in the datafile. Nevertheless, like any other data, these datafiles also need proper calibration.

- **Time Calibration**
  The time frames that are not part of the oscillation are excluded in this process. Lets say we have finite number of time frames $t_0, t_1, t_2, ...t_n$. If k number of frames are excluded from the beginning, then we have a new list of time, $t'_0, t'_1, t'_2, t'_3, ...t'_{n-k}$, where $t'_0 = 0$ sec and $t'_m = m \times (t_1 - t_0)$. We can find the time frames using this relation because the frame rate is constant throughout.

- **Height Calibration**
  The height of the liquid is calibrated using the formula for initial height (eq. (1)). From the equation we know the value of height of column at t=0. The rest can be done using the following relation,

  $$z_m = z_0 + (z'_m - z'_0)$$

  where z_0 is initial height and z'_m is the height of mth frame before calibration.

The calibration is done using python. The codes for which is submitted in the drive folder along with other codes.

Numerical Solution of the Models

The equations 4, 15 and 16 are then solved numerically with `scipy.integrate.odeint` package. The working of the package to solve differential equations numerically has been discussed in *Appendix B* [3]. After this step we have a list of heights for a corresponding input time list. However, to use the scipy DE solver, one must provide the value of damping coefficients.

Fitting of Damping Coefficients

The `scipy.optimize.curvefit` was not compatible with the `odeint` package. As a result we had to find out the best fit for b and Ω using a different method. We found out the goodness of the fit using a Chi-square method for a given value of damping coefficient. After that we iterate the value of the coefficient within a plausible range and find out the coefficient for which Chi-square value comes least. This method is further discussed in *Appendix C*. The formula for Chi-square value is given by,

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{21}$$

where O_i is the experimental value and E_i is the value we get by solving the model.

Fast Fourier Transform

The FFT has been done using the package `numpy.fft`. The usefulness of FFT and a brief overview of why it is used is discussed in *Appendix D*.

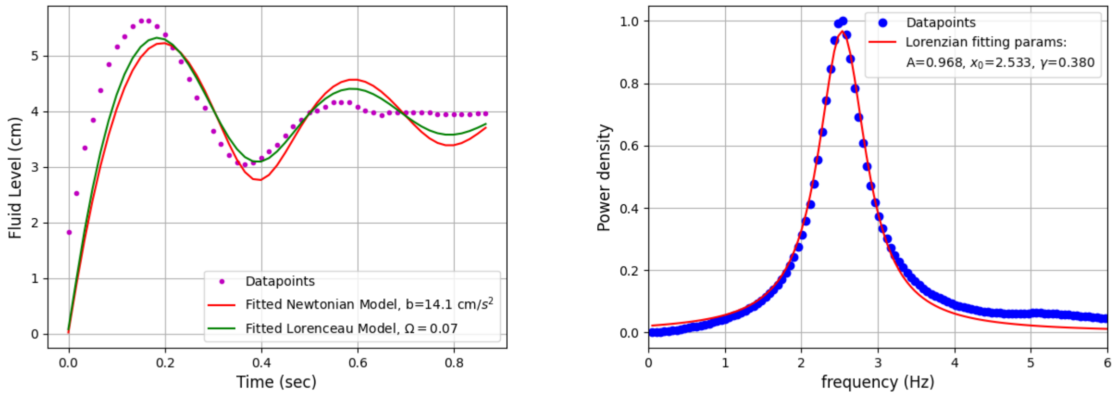
With the above mentioned methods, we get the best-fitted graphs for both the models and the graphs for power spectrum with lorentzian fit. The plots are shown here.

Results

Plastic Straw with water

r= 0.523 cm, H= 13.95 cm, $\rho = 1gm/cm^3$

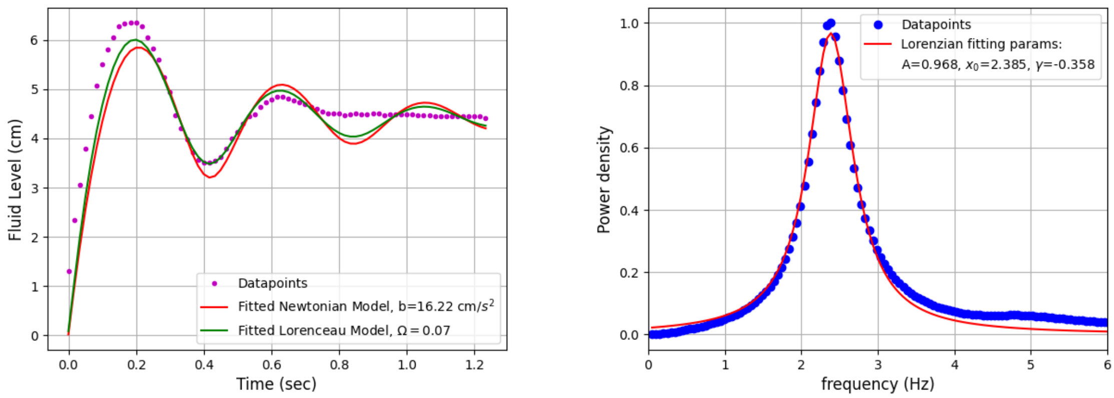
h= 3.9 cm,



(a) Data with Newtoian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 4: Parameters, b= 14.10 cm/s, b'= 12.11 s^{-1}, Ω= 0.23, f_0= 2.53 Hz

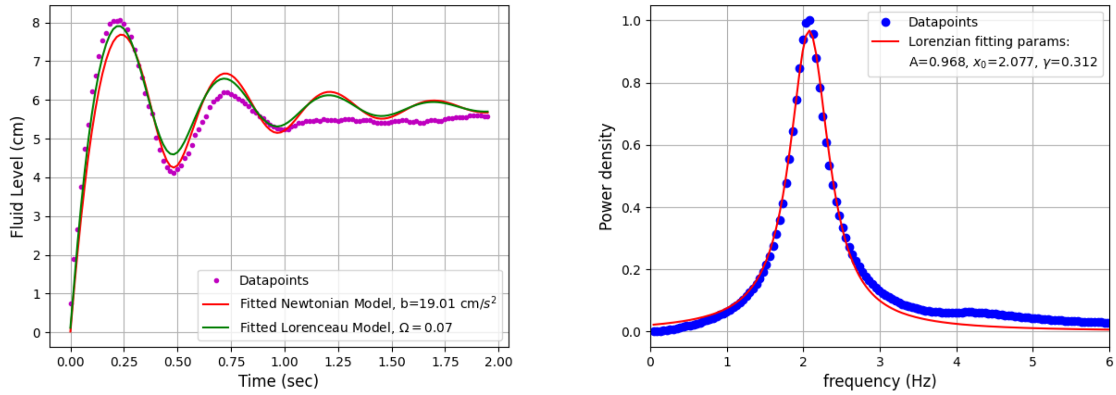
h= 4.4 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 5: Parameters, b= 16.21 cm/s, b'= 13.94 s^{-1}, Ω= 0.23, f_0= 2.38 Hz

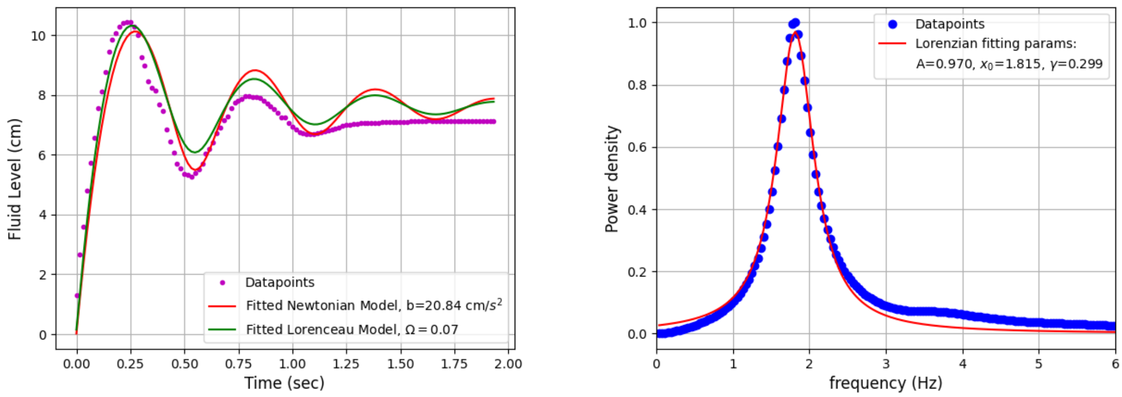
h= 5.8 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 6: Parameters, b= 19.01 cm/s, b'= 16.34 s^{-1}, Ω= 0.23, f_0= 2.08 Hz

h= 7.6 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 7: Parameters, b= 20.84 cm/s, b'= 17.91 s^{-1}, Ω= 0.24, f_0= 1.82 Hz

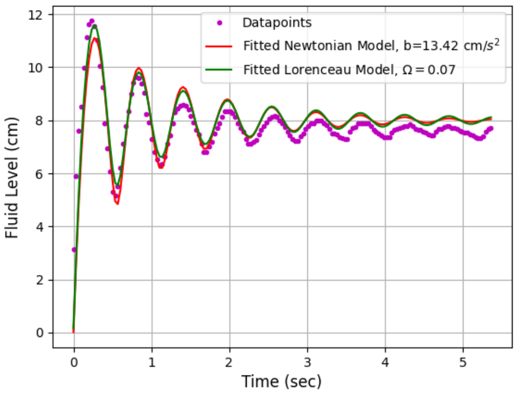
Table 1: The values are tabulated here

Sl. No.	h (cm)	b (cm/s)	b' (s^{-1})	Ω	Natural frequency(Hz)
1	3.9	14.10	12.11	0.23	2.53
2	4.4	16.21	13.94	0.23	2.38
3	5.8	19.01	16.34	0.24	2.08
4	7.6	20.84	17.91	0.23	2.82

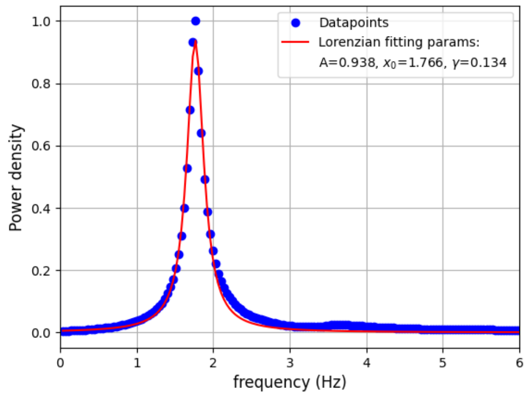
Glass Tube with water

r= 0.55 cm, H= 61.5 cm, $\rho = 1 gm/cm^3$

h= 8.0 cm,



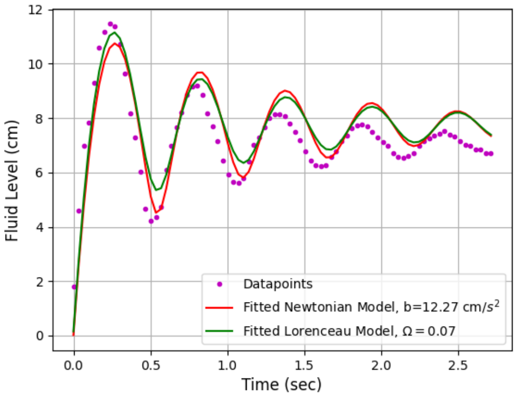
(a) Data with Newtonian and Lorenceau fitting



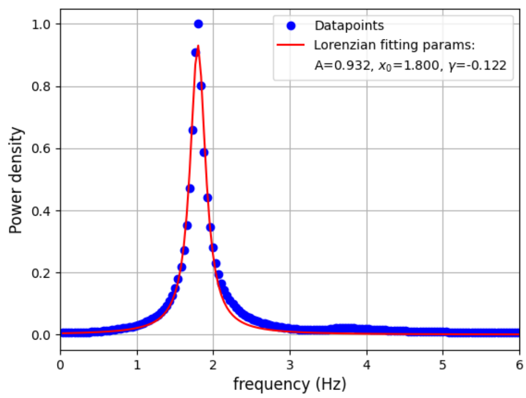
(b) Power Spectrum with Lorentzian Fitting

Figure 8: Parameters, b= 13.42 cm/s, b'= 12.76 s^{-1}, Ω= 0.08, f_0= 1.77 Hz

h= 7.7 cm,



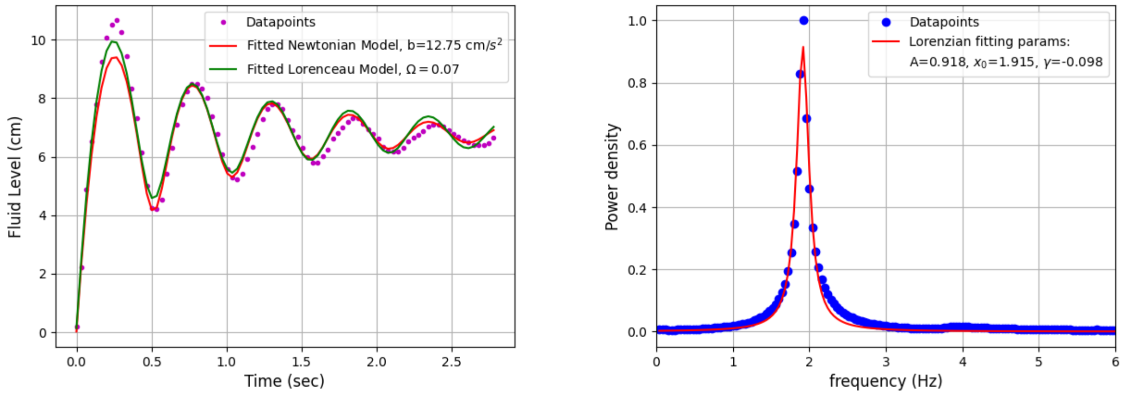
(a) Data with Newtonian and Lorenceau fitting



(b) Power Spectrum with Lorentzian Fitting

Figure 9: Parameters, b= 12.27 cm/s, b'= 11.66 s^{-1}, Ω= 0.08, f_0= 1.80 Hz

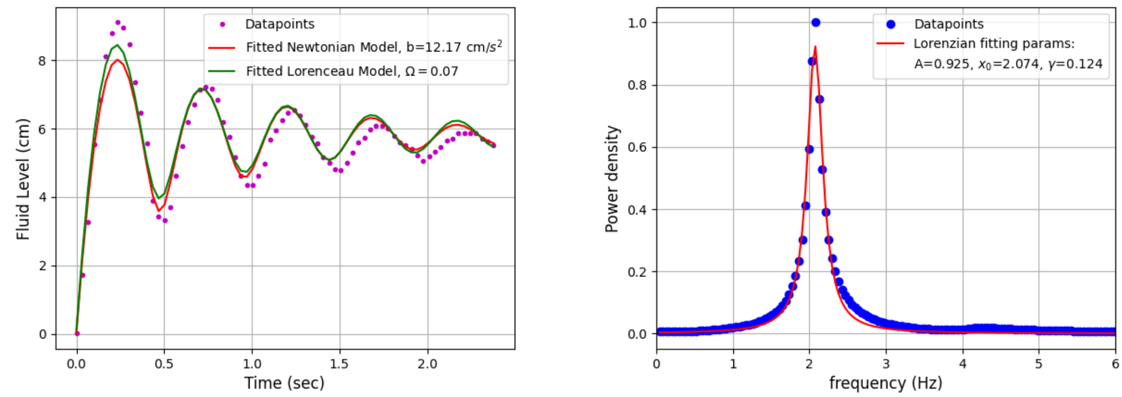
h= 6.8 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 10: Parameters, b= 12.74 cm/s, b'= 12.11 s^{-1}, Ω= 0.05, f_0= 1.92 Hz

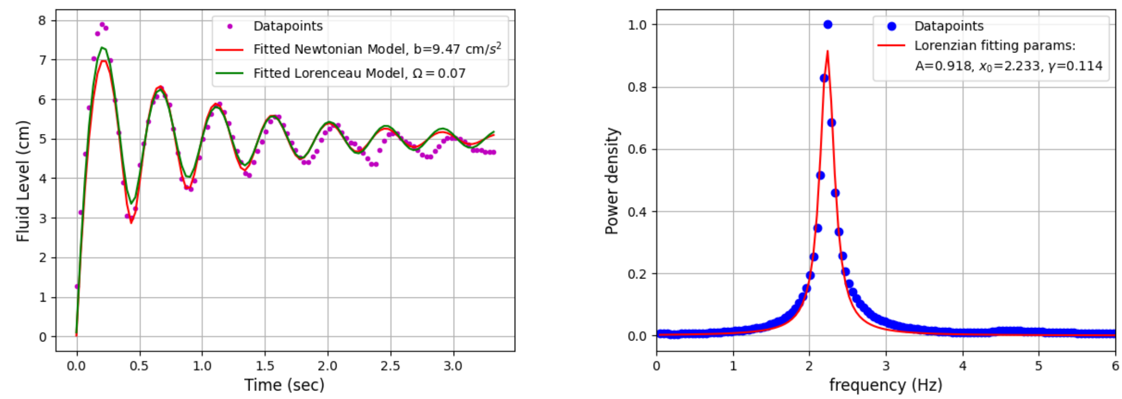
h= 5.8 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 11: Parameters, b= 12.17 cm/s, b'= 11.56 s^{-1}, Ω= 0.07, f_0= 2.07 Hz

h= 5.0 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 12: Parameters, b= 9.47 cm/s, b'= 9.00 s^{-1}, Ω= 0.05, f_0= 2.23 Hz

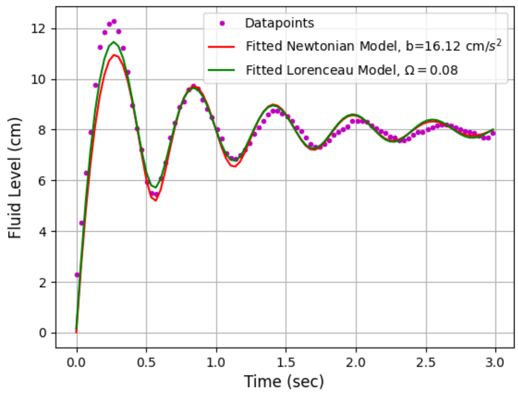
Table 2: The values are tabulated here

Sl. No.	h (cm)	b (cm/s)	b' (s^{-1})	Ω	Natural frequency(Hz)
1	8.0	13.42	12.76	0.08	1.77
2	7.7	12.27	11.66	0.07	1.80
3	6.8	12.74	12.11	0.05	1.92
4	5.8	12.17	11.56	0.07	2.07
5	5.0	9.47	9.00	0.05	2.23

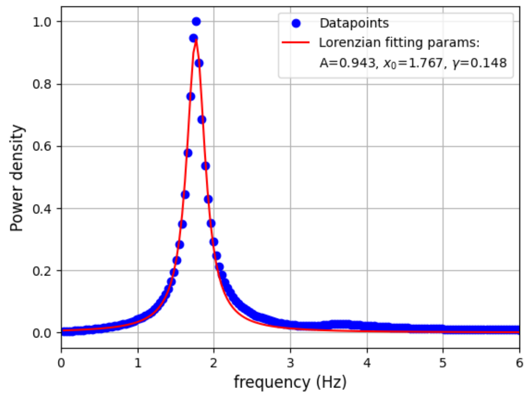
Glass Tube with Alcohol Solution

r= 0.55 cm, H= 61.5 cm, $\rho = 0.894 gm/cm^3$

h= 8.0 cm,



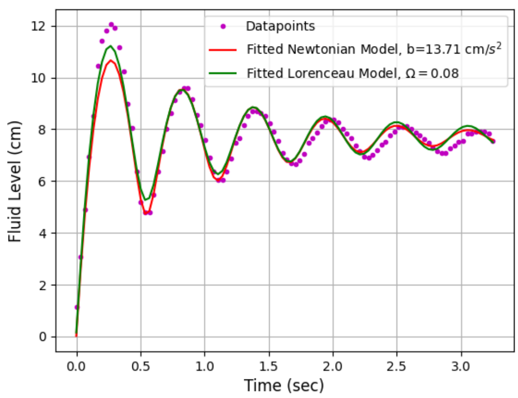
(a) Data with Newtonian and Lorenceau fitting



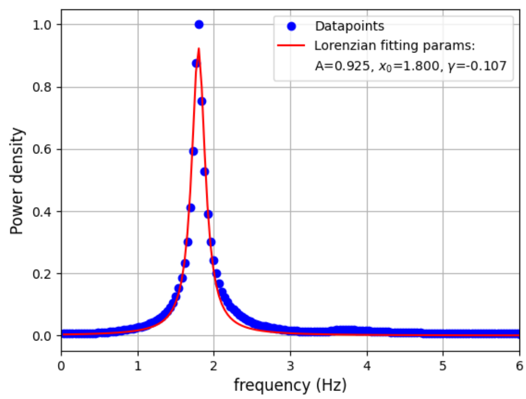
(b) Power Spectrum with Lorentzian Fitting

Figure 13: Parameters, b= 16.12 cm/s, b'= 13.70 s^{-1}, Ω= 0.08, f_0= 1.77 Hz

h= 7.7 cm,



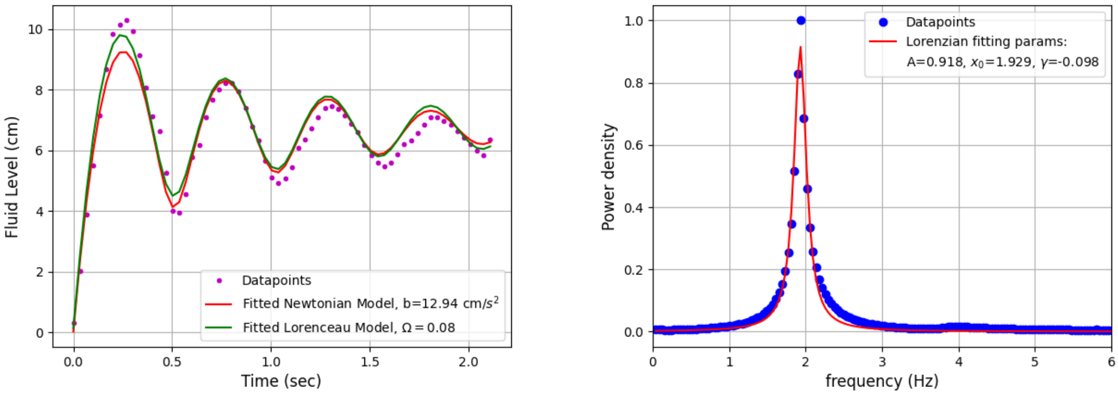
(a) Data with Newtonian and Lorenceau fitting



(b) Power Spectrum with Lorentzian Fitting

Figure 14: Parameters, b= 13.71 cm/s, b'= 11.65 s^{-1}, Ω= 0.07, f_0= 1.80 Hz

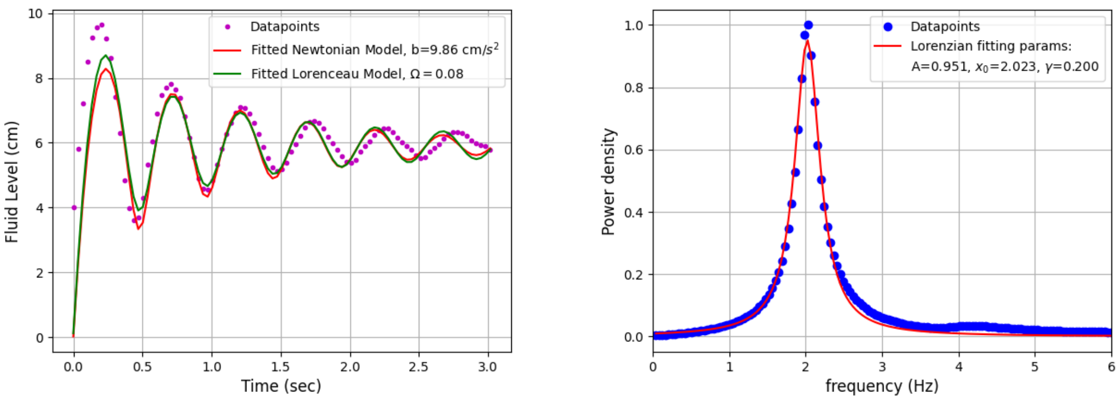
h= 6.7 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 15: Parameters, b= 12.93 cm/s, b'= 10.99 s^{-1}, Ω= 0.05, f_0= 1.93 Hz

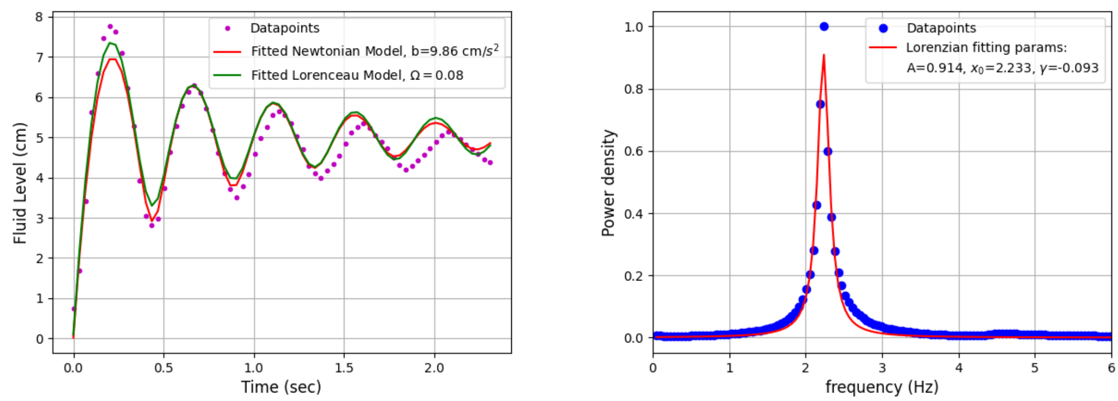
h= 5.9 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 16: Parameters, b= 12.46 cm/s, b'= 10.58 s^{-1}, Ω= 0.08, f_0= 2.02 Hz

h= 5.0 cm,



(a) Data with Newtonian and Lorenceau fitting (b) Power Spectrum with Lorentzian Fitting

Figure 17: Parameters, b= 9.85 cm/s, b'= 8.37 s^{-1}, Ω= 0.05, f_0= 2.23 Hz

Table 3: The values are tabulated here

Sl. No.	h (cm)	b (cm/s)	b' (s^{-1})	Ω	Natural frequency(Hz)
1	8.0	16.12	13.70	0.08	1.77
2	7.7	13.71	11.65	0.07	1.80
3	6.7	12.93	12.99	0.05	1.93
4	5.9	12.46	10.58	0.08	2.02
5	5.0	9.85	8.37	0.05	2.23

Relation Between h and b'

From the three different combination of liquid and pipe material, h and b' are plotted and are shown here.

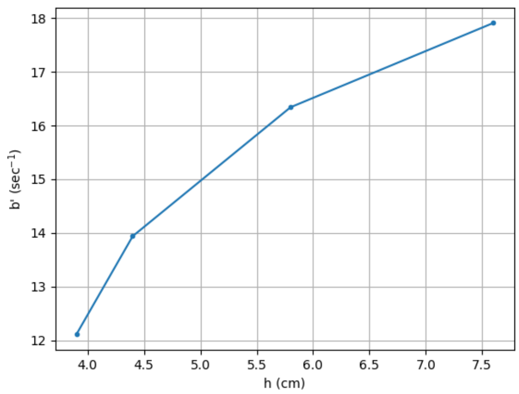


Figure 18: Graph between h and b' for water in plastic straw

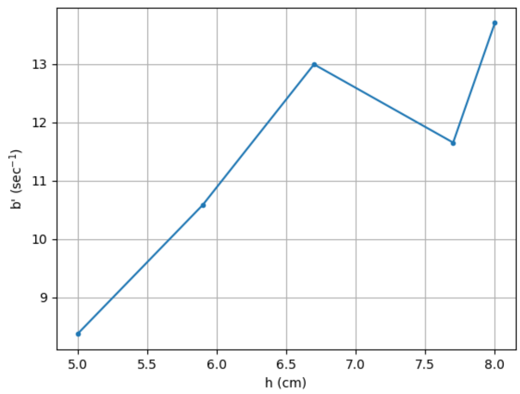


Figure 20: Graph between h and b' for alcohol solution in glass tube

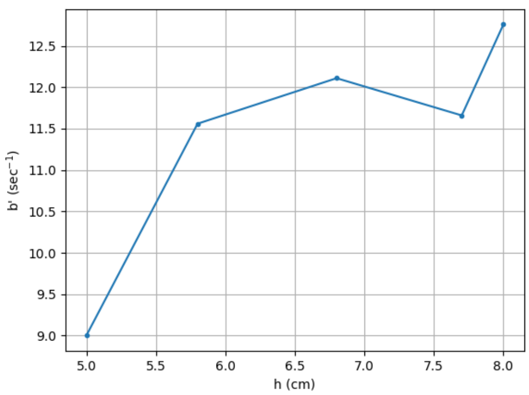


Figure 19: Graph between h and b' for water in glass tube

From the figures above (18, 19, 20), we can safely conclude that the damping coefficient increases when the height of pipe inside fluid level increases. However, we cannot form a relation between h and b' that may further deduce the pattern of the curve from this many datapoints. We need more number of datapoints for that purpose.

Estimation of g

Now we plot the f_0^2 vs (1/h) straight line curve with slope ($g/4\pi^2$) (eq. 20).

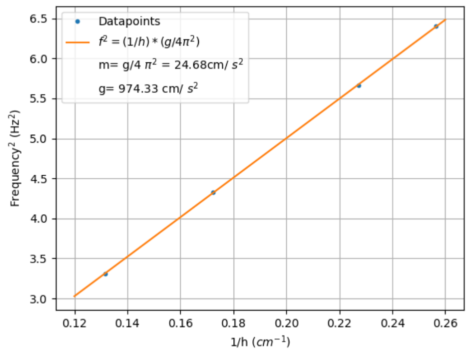


Figure 21: f_0^2 vs (1/h) for water in plastic straw

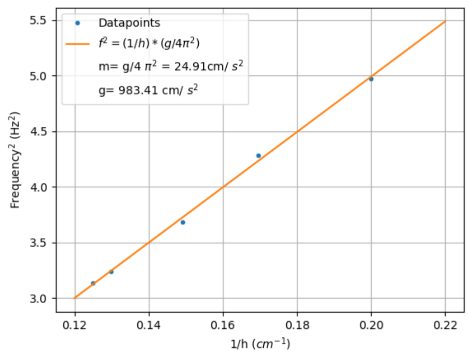


Figure 22: f_0^2 vs (1/h) for water in glass tube

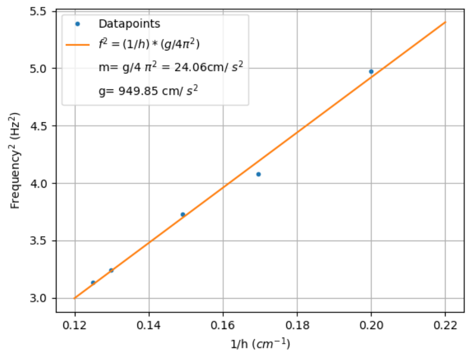


Figure 23: f_0^2 vs (1/h) for alcohol solution in glass tube

From the figures 21, 22, 23, we get value of g as 974.33 cm/s^2, 983.41 cm/s^2 and 949.85 cm/s^2 respectively. This is in close proximity of the actual value of g. This proves that the Fast Fourier Transform was a success and the system follows Hooke's Law (eq. 17) for small oscillations.

Summary and Discussion

Accuracy of Data

As it can be seen, the accuracy of data is much less than expected. The dataset deviates highly from the models. This is not an error due to fitting, as this issue doesn't resolve when the fitting is done using a slider. Rather, this rises from the experiment. In most of the datasets, the first node of oscillation shows higher amplitude than the models. This might be a result of less number of time frames at the first moment of the experiment, when the fluid starts to rise. The sources of error are thoroughly discussed in the next section.

Comparison between models

Both of the models predicts the system quite closely, however, the Lorenceau Model gives a better approximation for the first amplitude. The average goodness (Chi-square) of the fitting is also better for Lorenceau model. So, it is safe to conclude that is Lorenceau model is a better approach than the Newtonian one. That is why the Lorenceau Model is used for the power spectrum.

Comments on b'

The value of b' determines the damping of the oscillation, in other words, it determines how long we can observe the oscillations. In case of plastic straw, the friction between straw and fluid increases significantly, giving rise to the value of b' (12-18 s^{-1}). Whereas the value is much less in case of glass pipe (8-12 s^{-1}). That is why we can observe only two oscillations with plastic tube, but more oscillations with glass tube.

Also, from figures 18, 19 and 20, we can deduce that the value of b' increases with increasing h. Although, to get a quantitative relation more experiments should be done with more datapoints, different liquids and different pipe materials.

Comments on Ω

The value of Poiseuille friction (Ω) has been found out to be 0.23 in case of plastic tube and 0.07 in case of glass tube. However, it should depend on fluid instead of the material. The reason behind this is that the model was based on the assumption that it was laminar flow. But in reality it is a mixture of laminar and turbulent flow.

Also, as equation 14 suggests, Ω has no dependency on h.

Comments on the Power Spectrum

The goodness percentage of the power spectrum fitting with lorentzian function is really high. This proves that the system follows the Hooke's Law for small oscillations. The calculated natural frequency of the system also matches with the derived natural frequency. It does not depend on fluid nature or the material of pipe. The calculation of g from this data further supports the claim.

Sources of Error

Better results could have been produced using proper laboratory set up. The probable sources of error are:

- The frame rate of the video was not sufficient. Sometimes the video was too blurry to track the height of the fluid.

- The release of pressure at the open-end of the pipe was not instantaneous. As a result, there was ambiguity in starting point of the oscillation. It does not happen instantaneously, which is a huge factor for the fitting.

- The pipe was held by hands, not stand and clamp. As a result small vibrations and tilts added up to the error bar.

Future Scopes

This experiment is almost at a dead-end. However, one can take better data using better camera and different fluids to get better fitting. An approach to find a quantitative relation between h and b' can also be taken.

Acknowledgements

On a personal note, I learnt a lot of tools like odeint, fft etc. while doing this experiment. I would like to thank Dr. Pratap Kumar Sahoo and Dr. Gunda Santosh Babu for letting me choose this experiment and helping with the valuable suggestions. Dr. Pratap Kumar Sahoo also let us use the glass pipe from his lab, without which it would have been difficult to complete the experiment. I would like to thank SPS for this opportunity and First year lab, SCS for letting us use some of their lab equipment.

I am grateful Tamoghna for helping me take the data and completing the data collection while no one else from our group was not on campus. I would also like to thank Dwaipayan, Sourav and Tamoghna for the important and useful discussions we had.

I would like to thank Dr. Smith for sharing the python codes of their work [4], which helped a lot in the modelling. I would also like to extend my gratitude towards the tracker development team for their free software

Appendix A: Links

The videos, tracked datafiles and necessary python scripts can be found here: `https://drive.google.com/drive/folders/15ireubu8UAl0MnhBZCov3KoObrOEvcsK?usp=sharing`.

Appendix B: Numerical Fitting

The complexity of the working formulae of both the models makes it impossible to solve the equations analytically, thus we had to resolve to numerical fitting. The numerical fitting has been done by the python package `scipy.integrate.odeint`. The working of the module has been discussed thoroughly in *Hindmarsh and Petzold, 1995* [3].

The general approach to solve an ODE numerically is to take an initial condition and get the solution by gradually increasing the step size (h) and calculate the dependant variables by calculating some sort of slope or something with the step size. General numerical methods like Euler method, Runge-Kutta methods work in that way. However, to understand if we can take this approach, we need to understand something called a stiffness of an equation. In the simplest terms, a system of ordinary differential equation is said to be stiff if it has a strongly damped, or "superstable" mode. We can refer to the figure 24.

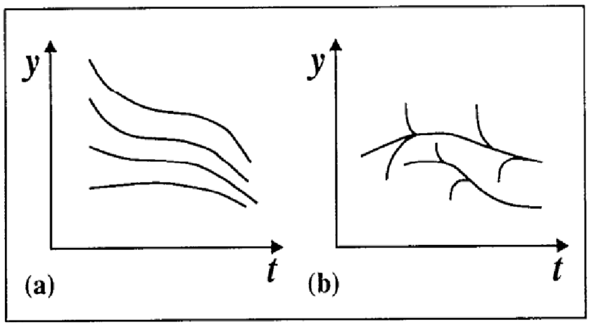


Figure 24: (a) Solutions to a non-stiff system vs (b) Solutions to a stiff-system.

For a non-stiff system of equation, the solutions with taking different h tends to merge at some time, but not very rapidly. The more the frequency of h, the closer the solution gets to analytical solution. The example of which is given in Fig. 24 (a), where an example of stiff systems is given in Fig. 24 (b). Here, the curves merge rapidly to a set of smoother curves, the deviation from the smooth curve being strongly damped as t increases.

Stiffness in a system of ODES corresponds to a strongly static behavior of the physical system being modeled. The stiffer the system, the lesser the chance of getting a good result using explicit methods (eg. Runge-Kutta methods). for those cases, methods that follow Implicit Euler Method is used. For an explicit method, if we know the solution at t=n-1 (initial condition), we can find out

$$y(t_n) = y(t_{n-1}) + h_n f(t_{n-1}, y_{n-1})$$

. In case of explicit methods the solution for t=n does not depend on t=n-1.

$$y(t_n) = y(t_{n-1}) + h_n f(t_n, y_n)$$

We can write this nonlinear system as,

$$F(u) = 0$$

This is typically solved by Newton iteration using Jacobian Matrix $\partial F/\partial u$.

Now, the `scipy.integrate.odeint` module works on an algorithm called LSODA, which does not depend on user to specify stiffness or step size. It automatically determines where the problem is stiff and chooses appropriate method, using dense or banded Jacobian Matrix.

Appendix C: Chi-Square Fitting

The Chi-Square fitting works on a more basic way than the usual `scipy.optimize.curvefit` method. The Chi-Square method actually calculates the goodness of the fit for a provided guess of the parameter.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the experimental value and E_i is the expected value.

Now, the algorithm that I wrote does not require a good initial guess. In fact, it iterates the values on a much wider range (which can also be provided) to find out the value of the parameter which gives the least value of Chi-Square, i.e. the one which deviates least from the observed value. In the next level of iteration, the result of the previous iteration is taken as guess and the process is repeated again to find a better and closer solution. This whole process iterated again and again until we get the solution upto desired decimal level.

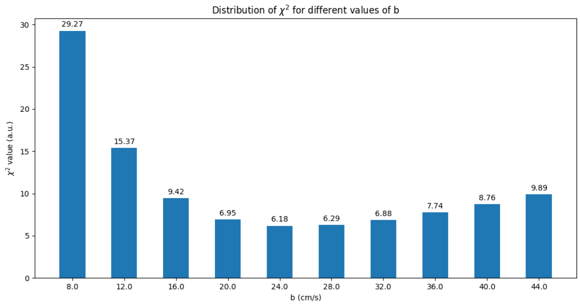


Figure 25: In this bar-plot of b vs χ^2, different values of χ^2 is shown for different guesses of b. With only this iteration, b=24 cm/s is the desired value. In the next step, we can further iterate it from b= 20 to 28 cm/s to find a closer approximation.

Appendix D: Fast Fourier Transformation

Fourier transform is a mathematical tool that decomposes functions depending on space or time into functions depending frequency. This is a very important tool for functional analysis, however, the formulae for that works on functions. To apply fourier transform on discrete dataset from an experiment, we need to use Discrete Fourier Transform (DFT). The formula for which is given by [5],

$$X_k = \sum_{n=0}^{N-1} X_n \cdot e^{-\frac{i2\pi kn}{N}}$$

taking $(\frac{i2\pi kn}{N}) = \omega_n$ simplifies the equation to

$$X_k = X_0 \cdot e^{-\omega_0 i} + X_1 \cdot e^{-\omega_1 i} + \cdots + X_n \cdot e^{-\omega_n i}$$

In matrix form,

$$\begin{bmatrix} F_0 \\ F_1 \\ . \\ . \\ . \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & ..... & 1 \\ 1 & \omega_n^2 & ..... & \omega_n^{n-1} \\ 1 & \omega_n^2 & ..... & \omega_n^{2(n-1)} \\ . & . & ..... & . \\ . & . & ..... & . \\ 1 & \omega_n^{n-1} & ..... & \omega_n^{(n-1)^2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ . \\ . \\ . \\ f_{n-1} \end{bmatrix} \tag{22}$$

$$F = X.f$$

where we get transformed F by multiplying the dataset f with a $N \times N$ matrix. This is also a computationally faster method than normal FT. However, calculation speed of this also depends on the order of N^2.

- `numpy.fft`

To make the process even faster, we use fast fourier transformation. There are different ways to do that. The one numpy follows goes like this. It transforms the $N \times N$ matrix with a multiplication of two matrices of order (N-1), one of which is a diagonal matrix. After that the other matrix is written as a multiplication of two other matrices of order (N-2), one of which is diagonal. This process continues until we get multiplication of N number of matrices and the result is a product of the just $2 \times N^2$ terms. By order calculation, it can be shown that this method depends on order of $Nlog_2N$. For a higher N, $\mathcal{O}(Nlog_2N) << \mathcal{O}(N^2)$. This is how numpy reduces the time of FFT.

References

[1] É. Lorenceau, D. Quéré, J.-Y. Ollitrault, and C. Clanet, "Gravitational oscillations of a liquid column in a pipe," *Physics of Fluids*, vol. 14, no. 6, pp. 1985–1992, 2002.

[2] R. P. Smith and E. H. Matlis, "Gravity-driven fluid oscillations in a drinking straw," *American Journal of Science*, vol. 87, pp. 433–435, 2019.

[3] A. C. Hindmarsh and L. R. Petzold, "Algorithms and software for ordinary differential equations and differential-algebraic equations, part ii: Higher-order methods and software packages," *Computers in Physics*, vol. 9, no. 2, pp. 148–155, 1995.

[4] R. P. Smith, "gravity-driven oscillations of fluid in a straw," https://github.com/r31415smith/AdvLab/tree/master/fluid_osc, accessed 2021.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 2007.