Technical report: Exercise 2

Nhat Pham

For Problem 1, using Euler method for solving ODEs, it is found that according to the simulation, Baumgartner did break the sound barrier with a fall of duration 277 s and a maximum speed which reached $348\,\mathrm{m\,s^{-1}}$. For Problem 2, a Gaussian wavepacket was successfully simulated using the finite differences method to solve PDEs (in this case the wave equation) for $\gamma = 1$, and investigating $\gamma \neq 1$ shows unstable regions for $\gamma > 1$, while for $\gamma < 1$, the models exhibit wave behaviour but far from the accuracy of the model when $\gamma = 1$.

I. PROBLEM 1: FREEFALL WITH DRAG

A. Introduction

Problem 1 required solving ordinary differential equations (ODE) in the form of Newtonian equations of motion. These are second order ODEs. The situation is of Felix Baumgartner free-falling from a great height y_0, with air resistances taken into account. Some parameters were given an appropriate value or range of values, some had to be estimated. The drag coefficient of Baumgartner was set to 1.0, the cross sectional area $0.4\,\mathrm{m}^2$, and the mass $70\,\mathrm{kg}$.

Part (a) called for a simple evaluation of the position and speed of the person free-falling analytically. Part (b) called for the Euler method, which requires two first order equations to solve the second order Newtonian ODE. Part (c) asks to vary the air density, which was a constant ρ_0 in previous parts. Part (d) used the free-fall results from part (c), but required an additional calculation—the speed of sound, to examine whether or not Baumgartner would break the sound barrier.

B. Theory

The Euler method involves a set of iterations which only requires the initial position and speed, y_0 and $0\,\mathrm{m}$ respectively, a timestep δt is also defined, which separates each iteration (see Equations 1, 2, and 3)

$$v_{y,n+1} = v_{y,n} - \delta t \left(g + \frac{k}{m} |v_{y,n}| v_{y,n} \right) \tag{1}$$

$$y_{n+1} = y_n + \delta t \cdot v_{y,n} \tag{2}$$

$$t_{n+1} = t_n + \delta t \tag{3}$$

C. Method

Part (a) was straightforward since the analytical equations were given and evaluation required only an array of time.

For part (b), To calculate y_{n+1} requires δt, y_n and also v_n. This means the order of which values were filled would be t, v and then y. This all happens within a `for` loop, and vectorisation did not seem plausible since the current indexed value is calculated using the previous indexed value. To examine the effect on varying the timestep δt, the simulation was repeated for several timesteps between 0.1–7 s and to quantify the error, the standard deviation of the difference δy between the analytical trajectory and the numerical trajectory was calculated. A similar method was used for vertical speed. To examine the effect of the quantity k/m on the motion, the mass m is varied between 50–$80\,\mathrm{kg}$ and k here is kept constant.

For part (c), the drag factor k now becomes a function of y, $k(y)$. The consequence of this is that whereas before v_{n+1} did not depend on y, but since it depends on k, which is now a function of y, it also depends on y. The same method for part (b) is used, but now the drag factor has to also be calculated at each step, since the altitude changes every step.

For part (d), because the speed of sound v_s is a function of temperature T, and T is a function of y, v_s is a function of y, $v_s(y)$. Since the simulation is discrete, the speed of sound is calculated using the y values obtained from part (c). The Mach ratio is then calculated, which is the ratio v/v_s.

To test what affects the Mach number, jump height and drag coefficient was varied between 1000–$40\,000\,\mathrm{m}$ and 0.05–2.0 respectively, keeping all other quantities the same (using Baumgartner's scenario).

D. Results and discussions

For part (a), two plots were generated, one for position and one for speed (Figure 1). Additionally, the result arrays were trimmed when the body reaches the ground.

For part (b), two plots were generated, one for position and one for speed (see Figure 2). The numerical prediction got close to the actual trajectory, which is the analytical prediction. The body stopped at similar times (part (a) predicts

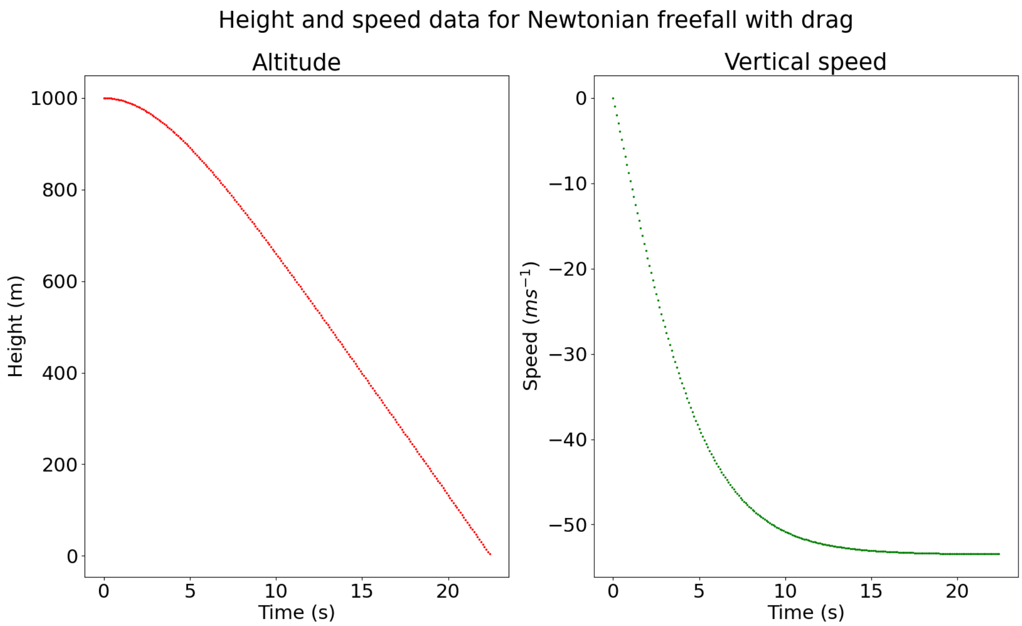

Figure 1. Analytical predictions of a free-falling body.

$t_f = 22.468\,\mathrm{s}$ while part (b) predicts $t_f = 22.400\,\mathrm{s}$).

On examining the effect of varying timestep, two plots were generated (see Figure 3). From both plots, it can be said that an increase in timestep increases the standard deviation in both position and speed. More interestingly, discontinuous features are seen in both plots, which makes the plots look like they consist of linear regions of different gradients stitched together, but with discontinuous jumps. The position plot (left) overall seems to be a quadratic increase in error. The speed plot (right) starts off appearing slightly quadratic but mostly linear, followed by a region where there exhibits no correlation (i.e. constant horizontal), this region is small and occurs roughly around $\delta t = 5\,\mathrm{s}$. After that, there is rapid increase similar to the position plot.

On examining the effect of varying k/m, the duration of the fall and the maximum speed achieved during the fall was plotted as a function of mass (see Figure 4). A larger mass seems to shorten fall duration, with a change that appears to be almost linear or slightly quadratic. A larger mass also increases the maximum speed at the order of $\sim 10^1$ $\mathrm{m\,s^{-1}}$ for the range of mass used, with a change that also appears to be almost linear or slightly quadratic.

For part (c), the value $y_0 = 39\,045\,\mathrm{m}$ was used here to match Baumgartner's initial conditions. Similar to (a) and (b), two plots were generated for position and speed (Figure 5). A minimum is observed in the speed versus time plot, which indicates that the body reaches a downward speed much higher than the terminal velocity. For reference, if the body were to fall without varying air density, its trajectory, with Baumgartner's initial conditions would look like Figure 6. The body would have taken more than twice as long to reach the ground, and its speed does not surpass the terminal velocity of $v \approx 55\,\mathrm{m\,s^{-1}}$. On the other hand, if the air density varies, then the downward speed reaches a magnitude of $\sim 10^2$. By varying the air density, the simulation is more realistic as it produces values closer to what actually happened.

For part (d), Baumgartner's fall reaches the maximum Mach number is 1.180 (to 3 d.p.), while the maximum speed is $348.030\,\mathrm{m\,s^{-1}}$ (to 3 d.p.). The plot for this is shown in Figure 7. There is a spike in the Mach ratio at around $50\,\mathrm{s}$, which seems correct since in 5, there is a spike in downwards speed at the same time. It appears that he would have broken the sound barrier, and this behaviour can be explained by the varying air density at different altitudes—without accounting for this, the jumper cannot reach a speed that great.

Three plots were generated for the maximum Mach number achieved (red), the maximum speed (green), and also the duration of the fall (blue). The Mach number increases with height, and appears to break the sound barrier (when it's larger than 1) above $30\,000\,\mathrm{m}$. This correlation appears to be linear for the most part, but at the start there is a lesser increase. The duration plot shows that increasing the height increases the duration loga-

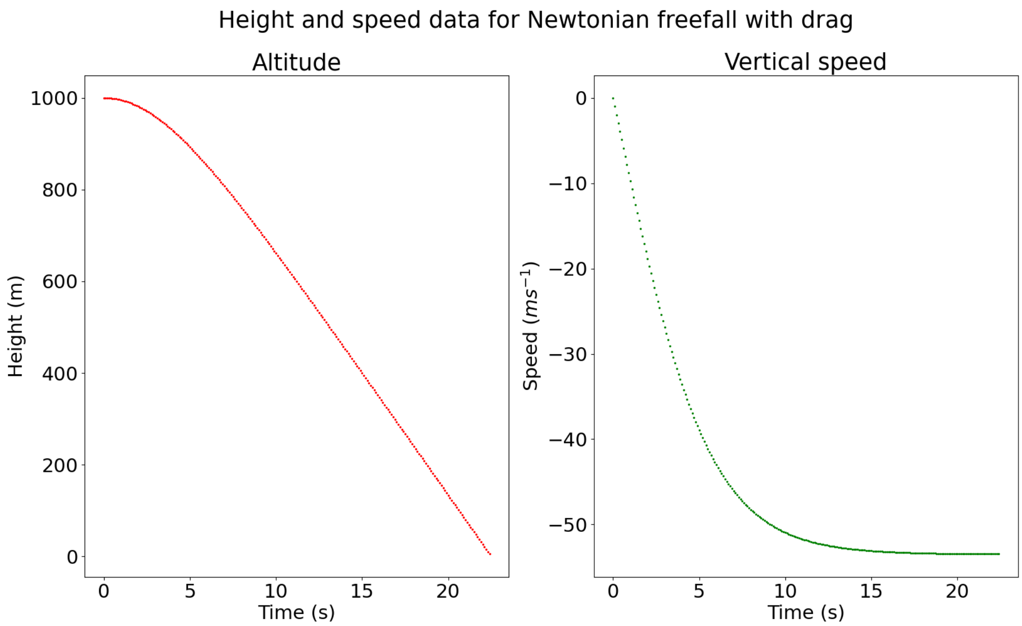

Figure 2. Numerical predictions of a free-falling body using Euler method.

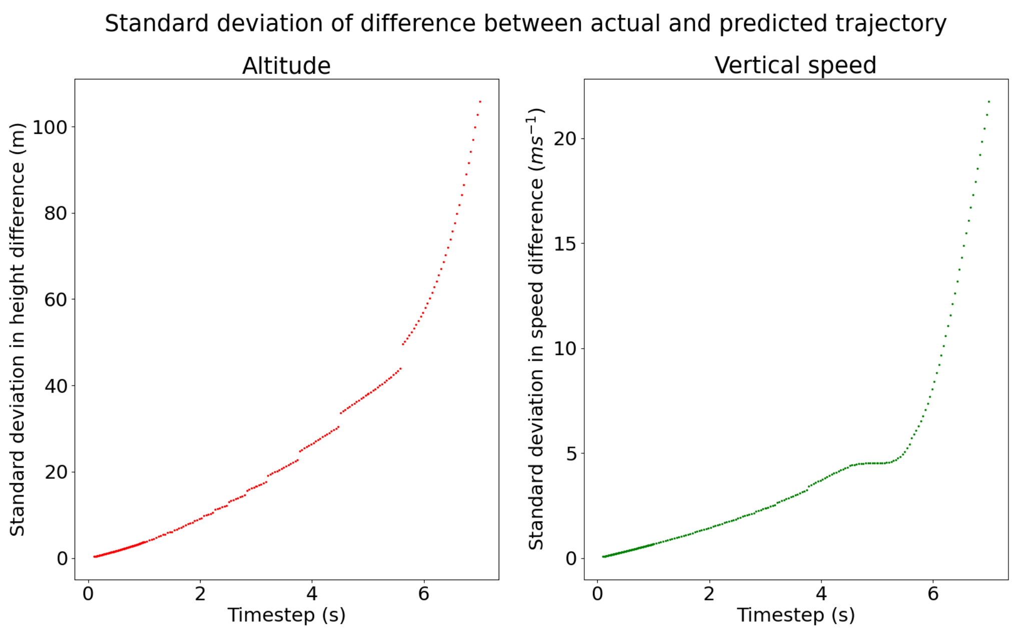

Figure 3. Investigating the effect of simulation timestep δt on the accuracy of the simulation

rithmically.

For jump height, three plots were likewise generated. As expected, increase the drag coefficient makes the falling object less aerodynamics, which results in a lower maximum speed achieved (middle green plot from Figure 9). However, it appears that the sound barrier can be breached even with a drag coefficient of ~ 1.75. The decrease in Mach ratio as drag coefficient increases appears to be exponential, while the duration increases logarithmically.

Overall, a timestep δt was chosen so that the standard deviation from the actual solution was small ($\delta t = 0.1\,\mathrm{s}$). Nevertheless, the final values were not exactly identical to values from the actual freefall. Baumgartner in reality fell for $259\,\mathrm{s}$ with maximum speed $373\,\mathrm{m\,s^{-1}}$ while in the simulation he was airborne for $277\,\mathrm{s}$ and reached $348\,\mathrm{m\,s^{-1}}$ (both values to 3 significant figures). The speed differed by a magnitude of $\sim 10^1\mathrm{m\,s^{-1}}$, or 6% of

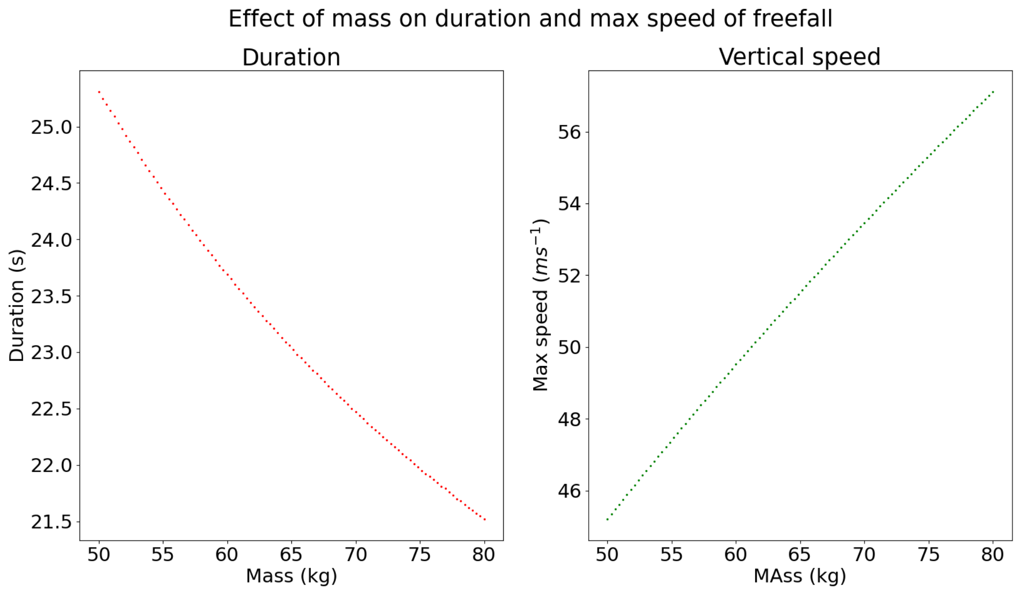

Figure 4. Investigating the effect of mass m of the body on duration and maximum speed of a freefall

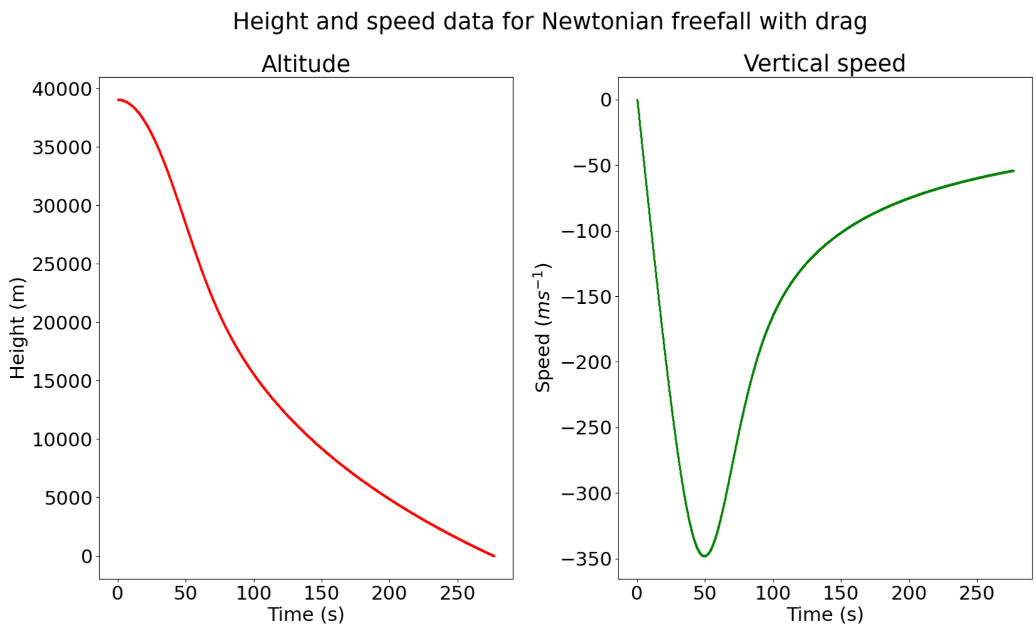

Figure 5. Numerical predictions of a free-falling body using Euler method, with air density varying as a function of position. Using Baumgartner's initial height

the actual measured speed. Thus, the accuracy in the result of part (c) or (d) (i.e. Figure 5 and 7) can be improved. If local truncation error was not small (in this case it was small), then the modified Euler method could have been considered. This requires an additional averaging of the slope from the original Euler method with the slope of the starting point to give a more accurate derivative. Nevertheless, improvements could be made by considering more physical factors beyond varying air density. In reality, how air turbulence works can be complicated and the motion of a body through air molecules can result in physics which might lead to slightly different results. For example, Baumgartner was tumbling around during his fall, so an average surface area was taken, but if an estimation of surface area for every step was used (as he was either falling down head first, or body first), this combined with varying air density would yield a more accurate model.

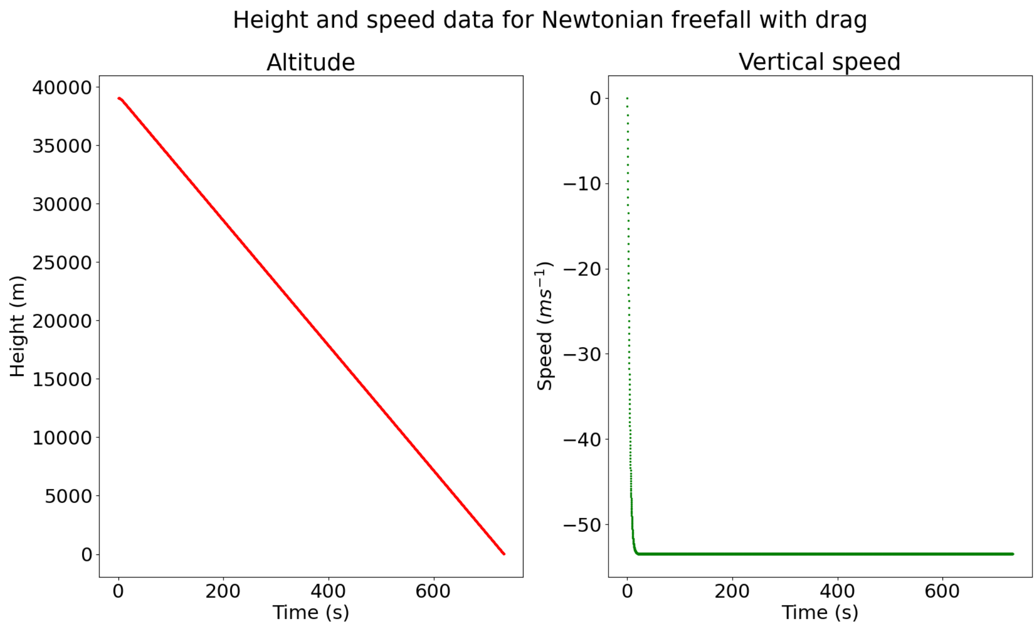

Figure 6. Numerical predictions of a free-falling body using Euler method. Using Baumgartner's initial height

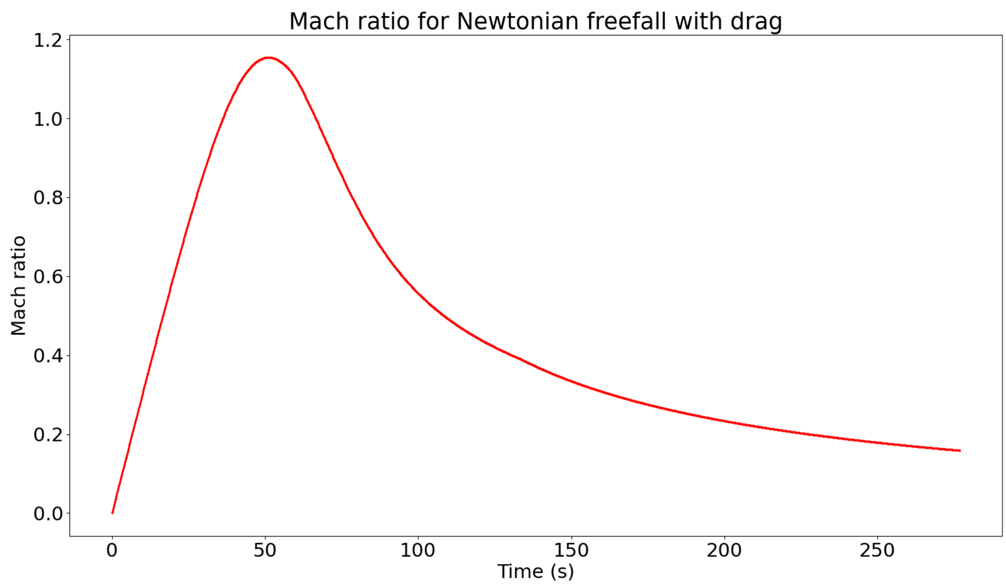

Figure 7. The ratio between Baumgartner's downwards speed and the speed of sound

E. Conclusion

Baumgartner did break the sound barrier according to the simulation, and the model and physical phenomena factored in did provide somewhat accurate results (with a maximum speed within 6% of actual maximum speed).

II. PROBLEM 2: WAVE EQUATION

A. Introduction

Problem 2 required modelling a Gaussian wave on a string, which here spans in the x direction. The position of the wave in the y direction at any given time t obeys the wave equation. Part (e) requires evaluating the solution of the partial differential equation (PDE), $u(x,t)$ numerically, and

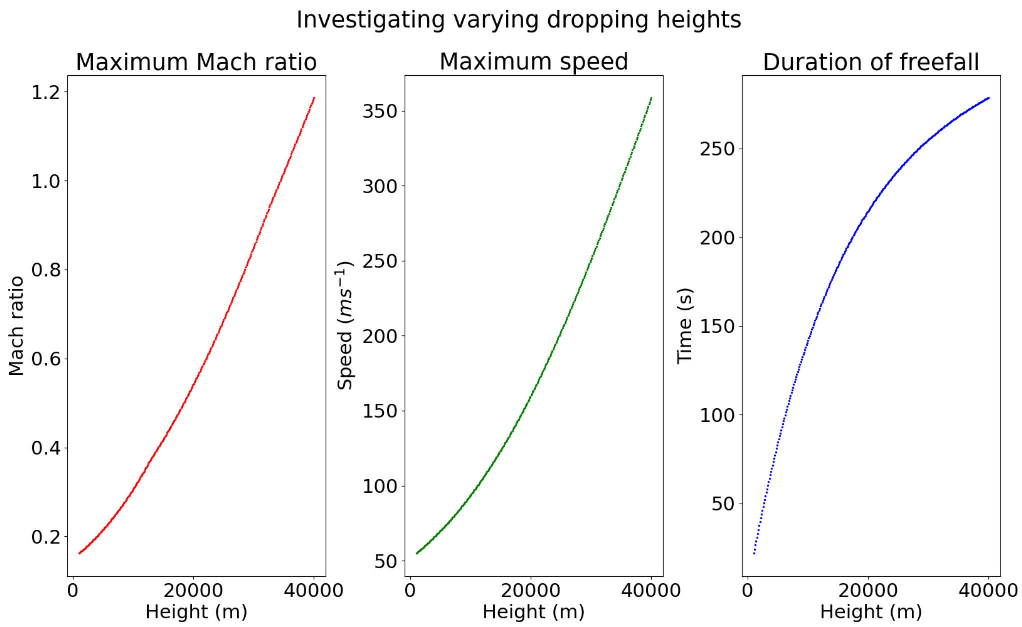

Figure 8. Investigating the effect of jump height y_0 on the maximum speed achieved during freefall

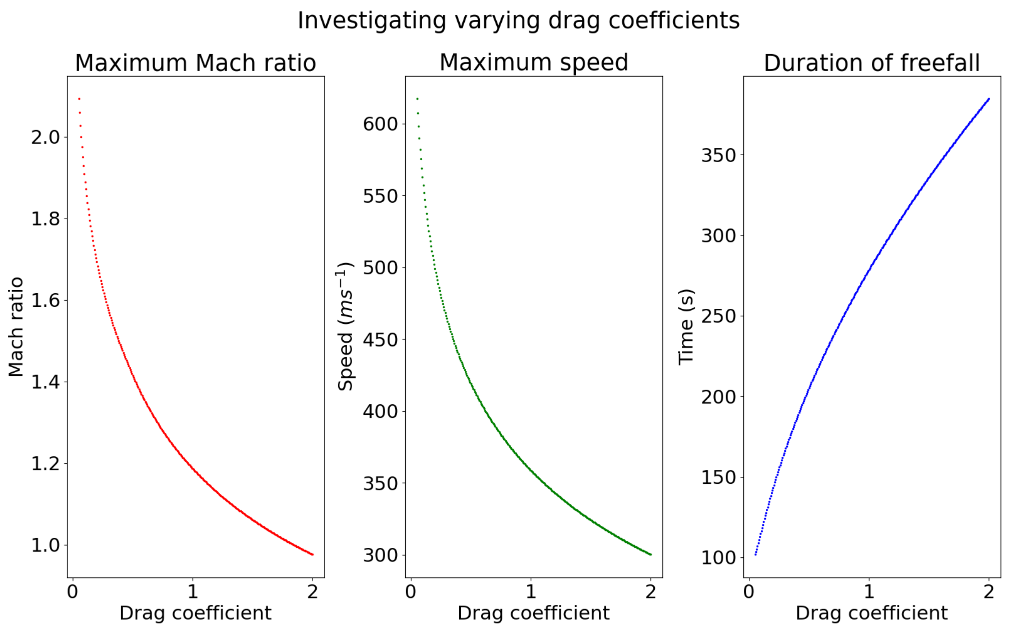

Figure 9. Investigating the effect of drag coefficient C_d on the maximum speed achieved during freefall

the finite differences method is used. Part (f) examines the effects of the model from part (e) for different values of γ.

B. Theory

The finite differences method was used to solve the wave equation for a 1-dimensional string (in the x direction) (Equation 4)

$$\begin{aligned} u(x, t+\tau) = \gamma[&u(x+q, t) \\ &+ u(x-h, t)] \\ &+ 2u(x,t)(1-\gamma) - u(x, t-\tau) \end{aligned} \tag{4}$$

With the zeroth timestep $u(x, 0)$ initialised by a Gaussian wavepacket, and the first timestep $u(x, \delta t)$ calculated using the central difference method (Equation 5).

$$u(x,\tau) = \frac{1}{2}[u(x+h,0) \\ u(x-h,0)] + \tau u'(x,0) \tag{5}$$

C. Method

For part(e), the finite differences method requires two initial timesteps, of which the first can be calculated using the Gaussian equation given, and the second using the Gaussian equation and its derivative. Sequential timesteps can then be computed using the finite differences formula. At each timestep, the positions of all points are plotted and saved as an image in the directory of the running program.

The calculation of the initial timesteps and the proceeding timesteps can all be vectorised with numpy arrays. In this numerical solution, the value $\gamma \approx 1$ gives the exact solution of the wave equation, but by fixing γ, this puts limits on the other properties of the wavepacket or the string. Since the length density μ is kept constant, the tension T in the string has to be tuned so that the phase velocity ($v = \sqrt{T/\mu}$) satisfies for $\gamma = 1$.

For part (f), the simulation from part (e) was repeated for values of gamma from 0.1–1.32. To measure the difference in the actual solution versus the model, the standard deviation of the difference between positions was computed for each timestep, and then an average was computed for all timestep so that each gamma value has a corresponding error value.

D. Results and discussions

For part (e), three timesteps of the wave is shown in Figure 10. The wavepacket moves to the right, hits the right node and reflects back with negative amplitude. These are all expected behaviours. The wave is seen to return to its original starting point after around $t = 6.10\,\mathrm{s}$. The wavepacket appears to have sensible motion that seems feasible. Its amplitude becomes negative when reflected at the boundary $x = 0$ and $x = L$, as expected from a realistic system. A way of improving the simulation would be to play around with values such as A_0 (the amplitude of the wave, which was normalised in this case), Gaussian width σ and even timestep δt and investigate their effects on the simulation accuracy.

For part (f), a plot was generated showcasing the error induced by varying γ (see Figure 11). Animations of waves for various γ were also generated and uploaded to the following link (`https://imgur.com/a/zbUIvHW`) (along with $\gamma = 1$).

It is seen that the standard deviation decreases linearly as γ approaches 1 on either side, except for the strange region around $\gamma \approx 0.99$ where the standard deviation increases suddenly before becoming 0 at $\gamma = 1$ (see Figure 12 for snapshots of $\gamma = 0.99$). This could happen possibly due to the simulation getting closer to the actual solution, so that it has a phase velocity quite close to the actual phase velocity, but since it deviates from the actual solution, but with more "motion", it can result to more deviation from the actual wave, as opposed to lower values of $\gamma < 0.99$ because those seem to have a phase velocity much lower (they are not moving to the right substantially, but staying almost stationary and generating waves by oscillating transversely, see Figure 13 for phase velocity very close to $0\,\mathrm{m\,s^{-1}}$ and Figure 14 for phase velocity non-zero but not quite fast enough). For γ below 1, it seems like the increase in error is linear as γ gets further from 1 and approaches 0.

For values of γ larger than 1, the error increases exponentially, reaching surprising orders of magnitude. It appears that γ should not be larger than 1, as wave motion behaves far from what is expected (starting from a string with no wavepacket at all and slowly a wave with increasing amplitude would form, which seems to continue to increase in amplitude, with no phase velocity observed).

Another way to quantise the error in varying γ besides the standard deviation in all x positions could be to measure the phase velocity v_p of the simulation and comparing this to the actual phase velocity (as a standard deviation).

Additionally, errors can be introduced in simulations of $\gamma \neq 1$ as a result of how the first timestep $t = \delta t$ is calculated. While the zeroth timestep comes straight from the Gaussian equation to initiate the wavepacket, the first one is found using the central difference method and relies on an approximation that is made when $\gamma = 1$. But when $\gamma > 1$, this approximation does not describe how the wave actually behaves, as seen in Figure 15. Thus, for $\gamma > 1$, the simulation appears unstable.

To improve the simulation and account for $\gamma \neq 1$, another way of approximating the timestep besides Equation 5 should be used so that the first timestep is more accurate.

E. Conclusion

A Gaussian wavepacket was simulated for $\gamma = 1$, and investigating $\gamma \neq 1$ shows unstable regions for $\gamma > 1$. For $\gamma < 1$, the models exhibit wave behaviour but far from the actual model because its phase velocity is less than the actual phase velocity, while the tension and length density remain the same.

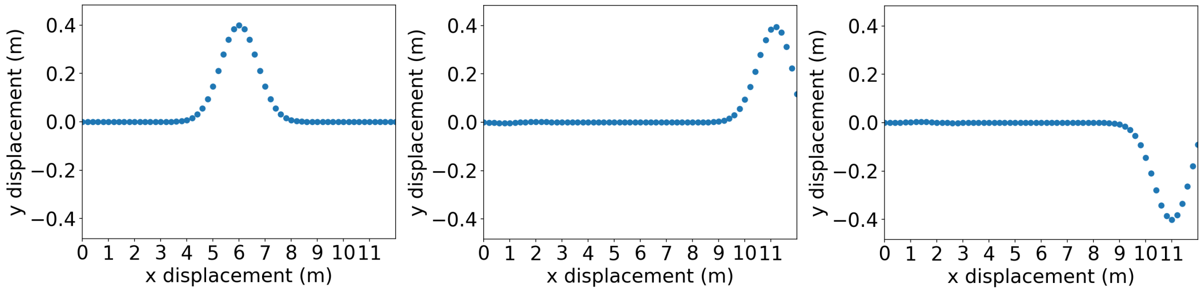

Figure 10. Wave propagation for a Gaussian wavepacket, $\gamma = 1$

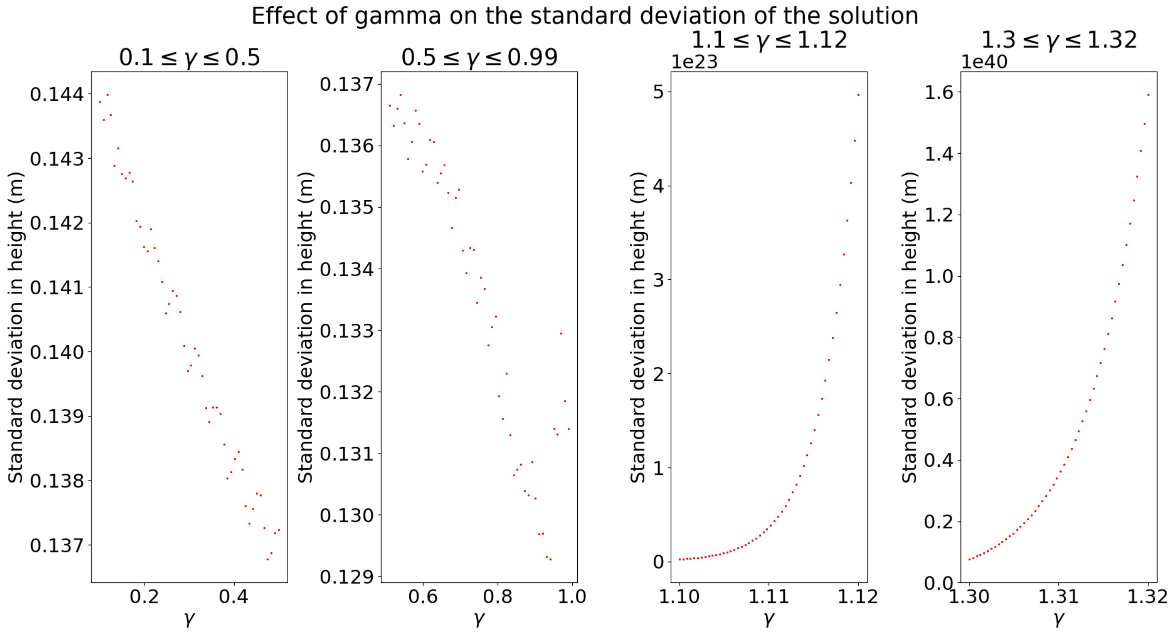

Figure 11. Effect of gamma on the standard deviation of the solution

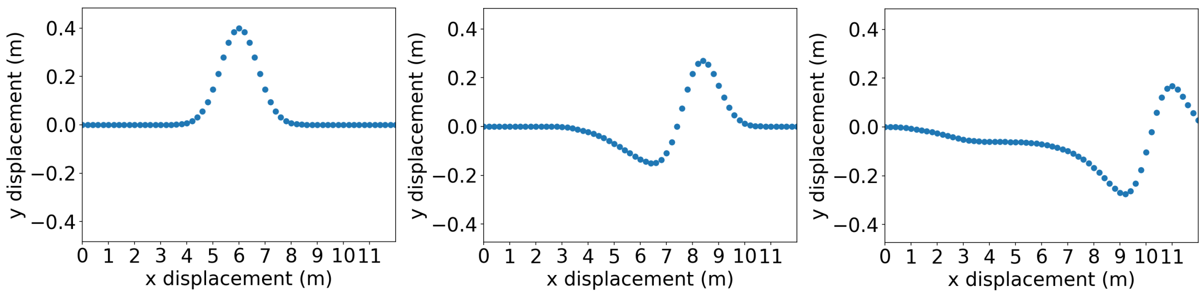

Figure 12. Wave propagation for a Gaussian wavepacket, $\gamma = 0.99$

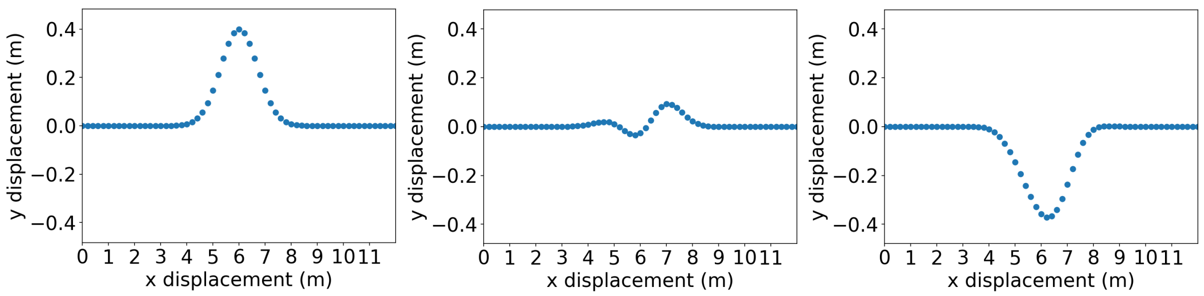

Figure 13. Wave propagation for a Gaussian wavepacket, $\gamma = 0.5$

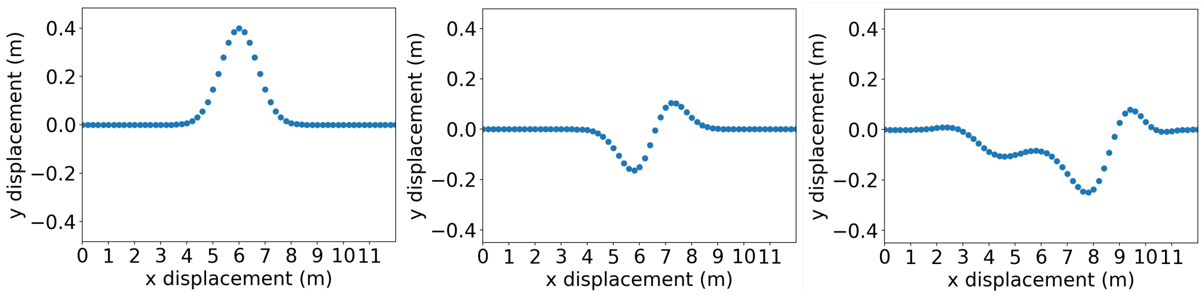

Figure 14. Wave propagation for a Gaussian wavepacket, $\gamma = 0.88$

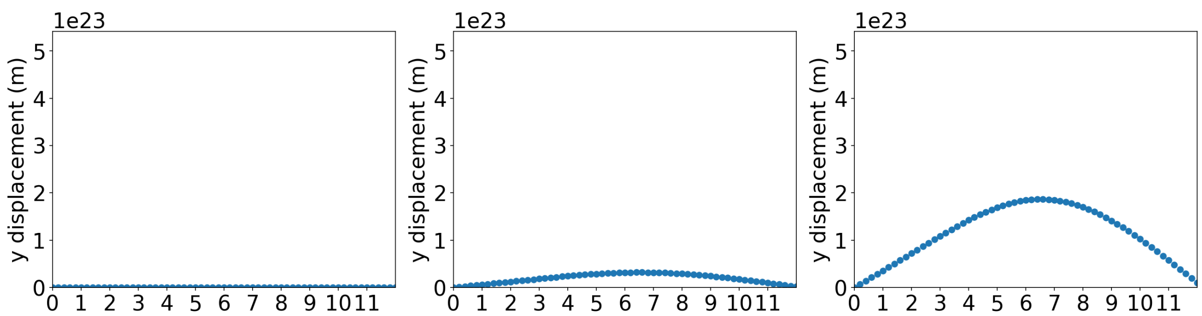

Figure 15. Wave propagation for a Gaussian wavepacket, $\gamma = 1.1$