Simulation of a 2D bouncing ball

Hannes Aspåker / aspaker@kth.se / 19950816-1131

January 21, 2020

1 Problem description

Situation: A hollow, elastic *two-dimensional* ball with a density (per unit length in the 3rd dimension) of 1kg/m is gradually filled with air until it reaches 4m in diameter in size. At time $t = 0$ it is dropped from a height of $y = 10$m, accelerated 9.82m/s^2 by gravity. At coordinates $y < 0$m there is a field imparting an upwards force of 900N per kilogram of mass to any part of the ball within the field, forcing the ball to bounce upwards towards its starting position.

Question: What is the relationship between the spring constant of the elastic material and the apex height of the ball's bounce? The height is defined by the position of the center of mass.

(I was given permission by the lecturer to work on a project not on the public list)

2 Model

The ball is modelled as a series of connected point masses (hereafter referred to as *nodes*). The various forces set upon the surface of the ball is applied to each node as a point mass, and the movement of the nodes then approximates the movement of the surface of the ball. Since the total mass of the ball is 1kg/m, the mass of each node is set as $\frac{1}{n}$kg/m, where n is the total number of nodes. The weight of the contained air is assumed to be negligible.

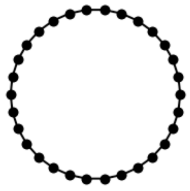

Figure 1: The ball is modeled as a series of connected point masses

A remark regarding units:

Since the ball is two-dimensional, it is modeled as the cross section of an infinitely long cylinder which is completely homogeneous in the 3rd dimension (depth). As

such, many units, constants and formulas will have an extra factor m^{-1} compared to their three-dimensional counterparts. This factor can be interpreted as "per meter of depth of the cylinder".

2.1 Elastic forces

The elasticity of the surface of the ball means that nodes will be pulled towards each other, with a force (per meter depth) that increases with separation distance. We model this using Hooke's law $F = kd$, where k is the spring constant (units $\mathrm{N/m}^2$) and d is the distance between the nodes in question (the circumference of the ball is assumed to be negligible in its relaxed state).

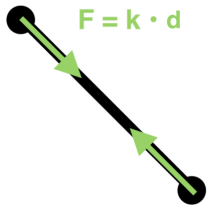

Figure 2: An elastic force $F = kd$ pulls nodes towards each other.

2.2 Pressure forces

We assume the ball is surrounded by vacuum and filled with a diatomic ideal gas. Using the properties of an ideal gas, the outwards pressure from the contained air on the surface of the ball can then be calculated as:

$$p = \frac{2}{5}\frac{E_I}{A}$$

where p is the outwards pressure, E_I is the internal energy of the gas (per meter depth) and A is the area of the ball (the area is calculated from the position of the nodes using the shoelace formula). The factor $\frac{2}{5}$ is associated with the dimensionless specific heat capacity of a diatomic gas.

When creating the ball, the internal energy is calculated from the spring constant and the desired radius of 2m, so that the pressure exactly counteracts the elastic forces at this radius.

The normal force (per meter depth) due to pressure on each segment of the surface is calculated as the pressure times the length of the segment. This force is then split between the two nodes sharing the segment.

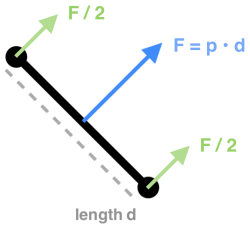

Figure 3: A force $F = pd$ due to pressure is applied normal to the surface.

2.3 Gravitational forces

A constant downward force (per meter depth) equivalent to an acceleration of 9.82m/s^2 is applied to each node to simulate gravity.

For every node at position $y = 0$ or below, an additional upward force equivalent to 900N/kg is applied to each node, corresponding to the field responsible for the bounce.

2.4 Damping forces

To simulate energy losses during deformation of the elastic material, a damping force (per meter depth) is applied in proportion to the velocity of each node, according to the formula:

$$\bar{F} = -m\gamma\bar{v}_r$$

where m is the mass of the node per meter depth, γ is the *damping constant* and $\bar{v}_r$ is the velocity of the node *relative to the center of mass* of the ball. We only dampen the velocity relative to the center of mass, because energy losses in the material should only appear when the nodes move relative to each other and deform the material, and not if they all move in tandem.

2.5 Rate of change of the internal energy

When the pressure of the gas expands the bubble, it performs work on the nodes and thus the internal energy decreases. In an analogous manner, when the area of the ball decreases, the nodes perform work on the gas and the internal energy increases. Here, we assume the contraction and expansion of the ball is an adiabatic process, i.e. it happens so fast that no heat is transferred to the environment and the temperature of the gas does not remain constant.

The work W done when the pressure force F_p moves a node a distance d in the direction of the force is given by $W = F_p d$. This leads to the following formula for the rate of change per second of the internal energy E_I (per meter depth):

$$\dot{E}_I = -\sum_i \bar{F}_{pi} \cdot \bar{v}_i$$

Where $\bar{F}_{pi}$ is the pressure force (per meter depth) on node i and $\bar{v}_i$ is its velocity.

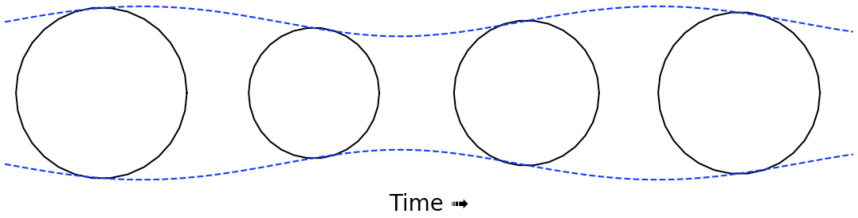

Figure 4: An example of a ball oscillating in size, with the horizontal axis representing the motion of time. The blue lines traces the size of the bubble over time.

3 Comparison of integration methods

Here, we will compare two common numerical methods for integrating Newtons equations of motion: Velocity Verlet and 4th order Runge-Kutta.

To compare these methods, we begin with a simple scenario: with no gravity or damping, each node is given an outwards velocity of 10m/s at $t = 0$. This causes to surface of the ball to oscillate, periodically increasing and decreasing in size (see fig 4). Because the damping constant γ is set to 0, the total energy should remain constant.

3.1 Velocity Verlet

Velocity Verlet is a method to solve systems of the form:

$$\begin{cases} \dot{x} = v \\ \dot{v} = a \end{cases}$$

However, this presents a problem when we consider the rate of change of the internal energy

$$\dot{E}_I = -\sum_i \bar{F}_{pi} \cdot \bar{v}_i \propto \sum_i a_i \cdot v_i$$

The derivative contains a product of *both* the acceleration a and the velocity v. This means it does not fit into the above system of equations, where the derivatives depend only on one or the other. I could not find a proper way to solve this, and the best solution available to me was to make a choice: either treat E_I it as a positional variable and update it in the same step as the positions x are updated, or treat it as a velocity variable and update it at the same step as the velocities v.

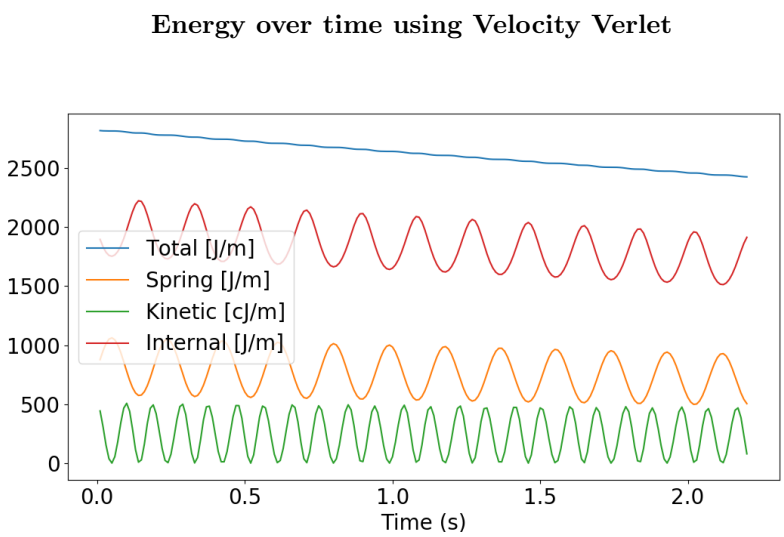

Figure 5: Various energies over time using Velocity Verlet. Here, a timestep $dt = 0.01$s and a spring constant $k = 300\text{N/m}^2$ were used. The kinetic energy has been scaled by a factor 100 to make it more visible.

Updating E_I at the same time as v resulted in singularities and wild fluctuations in energy. However, updating it at the same time as x resulted in a somewhat stable energy with only a slight drift (see figure 5). The slight drift shows that our addition of the internal energy to the system has broken the energy conservation of Velocity Verlet, and it is no longer symplectic. This is to be expected, as we've abused the method for something it was never meant to do.

3.2 4th order Runge-Kutta

Using Runge-Kutta, we yet again face the problem that E_I does not fit neatly into the system of equations, and yet again we must make the choice of updating it when the positions are updated or when the velocities are updated. Just like we saw in Velocity Verlet, updating E_I in the same step as the positions are updated in gives better results regarding energy conservation.

When compared to Velocity Verlet, we can clearly see that the Runge-Kutta method resulted in a drastically lower drift in energy for the same time step (see fig 6). Since this is a fourth order method, we should also see a faster reduction of the error when decreasing the time step compared to Velocity Verlet (which is only a second order method). Thus, 4th order Runga-Kutta is clearly the superior method for our use case. This method will be used for all subsequent analysis.

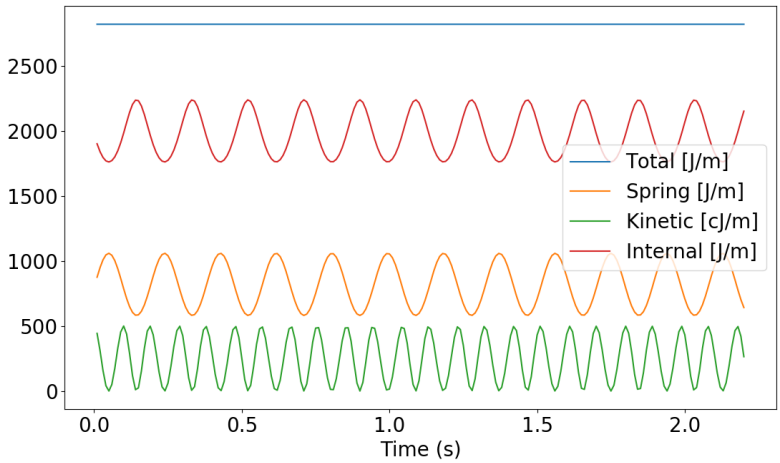

Figure 6: Various energies over time using 4th order Runge-Kutta. Here, a timestep $dt = 0.01$s and a spring constant $k = 300\text{N/m}^2$ were used. The kinetic energy has been scaled by a factor 100 to make it more visible.

3.3 Choosing a time step

To decide the size of the time step, which is a trade-off between speed of simulation and accuracy, we now introduce the scenario from the problem description: the ball is dropped from a height of 10m and bounces. To evaluate the effect of the size of the time step, we begin with no dampening so that the energy should remain constant.

Some experimentation showed that $dt = 0.001s$ gave good results, with reasonable speed and a loss of only 0.03% of the total energy after 5 seconds (see fig 8). By carefully studying the potential energy (purple line), it can also be seen that even with no damping the ball does not return to its initial height – some of the potential energy has been converted into vibrational kinetic energy of the ball's surface during the bounce.

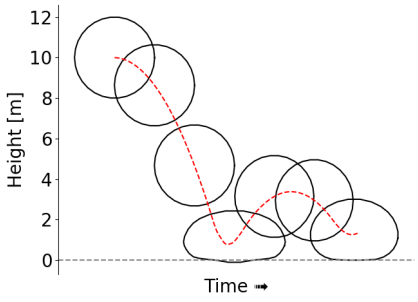

Figure 7: A visualization of a single bounce of the ball. The horizontal axis represents the advancement of time, and the red line traces the motion of the ball's center of mass.

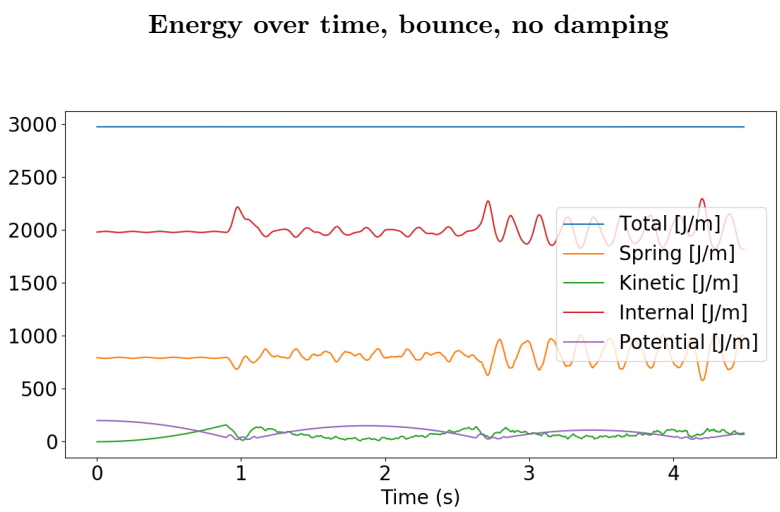

Figure 8: Various energies over time during a bounce, with no damping. Here, a timestep $dt = 0.001$s and a spring constant $k = 300\text{N/m}^2$ were used. For visual clarity, the potential energy of the elastic material (spring forces) has been separated from other potential energies (those from gravity and the lower force field).

4 Convergence

The simulation of the ball as a collection of point masses is an approximation. To know if we can trust the results of this approximation, it is prudent to check if the results converges to a specific value when the number of nodes increases and the surface approaches a continuous line of infinitesimal masses.

However, it is not as easy as simply increasing the number of nodes. The way the model is constructed, the interaction of the spring konstant k with the nodes depends on the angle θ between nodes. For example, using some simple geometry on a circular ball, it can be shown that the relationship between internal energy E_I and k at the desired radius of $2m$ is:

$$E_I = B \cdot \frac{k \sin \theta/4}{\cos \theta/2}$$

where B is a constant. Thus, to model the same physical scenario (with the same internal energy) when we increase the number of nodes from n_1 to n_2 and thus decrease the angle between nodes from θ_1 to θ_2, we need to also scale k by the factor:

$$\frac{\sin(\theta_1/4)\cos(\theta_2/2)}{\cos(\theta_1/2)\sin(\theta_2/4)} \approx \frac{\theta_1}{\theta_2} = \frac{n_2}{n_1} \quad \text{for small } \theta \text{ or large } n$$

However, this means that k will not converge towards a "true value" as $n \to \infty$, but rather grow towards infinity. This could represent a fatal flaw in our method, as the "true" continuous surface should not have an infinitely large spring constant. If the method is not a valid approximation, then the results of the simulation cannot be trusted. Unfortunately I stumbled upon this problem too late in the process to to anything about it.

If we are lucky, it is not a fatal flaw in the method, and instead it simply means that one of our assumptions (such as Hooke's law) is not valid at the smallest scales, but still is an acceptable approximation at larger distances. But at this time I am not able to judge the validity of this.

Still, we can at least show that our method converges in such a way that our results with this model should not significantly depend upon the number of nodes (see figure 9). In this example, we see that the height of the return bounce converges quite quickly, and only varies slightly for $n > 60$. Since the bounce height is the same for different (large) values of n, and k is only scaled by a constant factor between two different values of n, the relationship between k and the apex height of the bounce should remain the same for different values of n except for a constant factor.

The amount of nodes chosen for this report was $n = 30$ (30 nodes). This is not the best choice, as the results clearly have not converged fully at $n = 30$. However, it was not possible to choose a larger value due to time restraints, and the value at $n = 30$ is still within 3% of the value at $n = 120$.

5 Results & Discussion

For three values of gamma ($\gamma = 3.75\text{s}^{-1}, 7.5\text{s}^{-1}, 15\text{s}^{-1}$), 42 simulations were run for k-values between 100N/m^2 and 2200N/m^2 with a step size of 50N/m^2. For each simulation, the apex height of the bounce (as a % of the initial height of 10m) were noted and plotted in figure 10.

We can see that in the region $k < 620\text{N/m}^2$, the relationship between k and bounce height is approximately linear for all γ studied. In this region, lesser damping leads to higher bounces, as expected.

At around $k = 620\text{N/m}^2$, there is an inflection point at each line, were the rate of change starts to flatten out or even reverse. At exactly $k = 997\text{N/m}^2$, the bounce height is independent of γ.

The region $1000\text{N/m}^2 < k < 1500\text{N/m}^2$ is interesting, as the relationship between γ and height is reversed here: a higher γ results in a higher bounce!

Bounce height as a function of number of nodes (point masses)

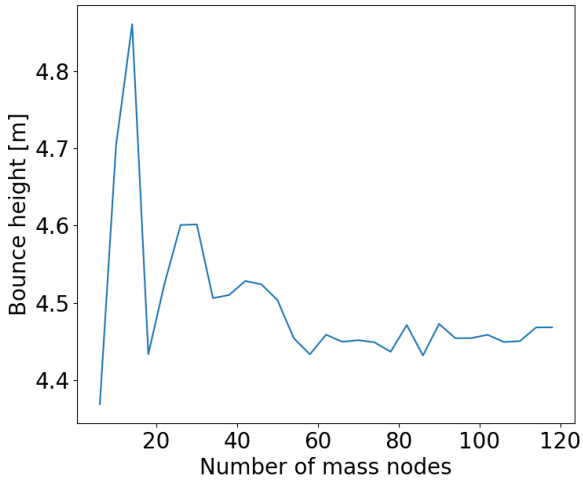

Figure 9: Apex height of bounce as a function of the amount of nodes n, with $dt = 0.001, \gamma = 6\text{s}^{-1}$. The initial value of k used for 6 nodes was $k = 30\text{N/m}^2$, which was then scaled for each n using the scaling factor previously discussed.

A possible explanation for this could be that – in this region – potential gravitational energy is converted to vibrational energy, lessening the bounce. A higher damping constant dampens these vibrations, preserving the height. A hypothesis for why it's more noticeable in this particular region could be that in this region the ball is affected by a $resonance frequency$. Figure 11 and 12 could support this hypothesis, as the oscillations in the energy from elastic forces oscillates at a very regular pattern in this region ($k = 1250\text{N/m}^2$), compared to the oscillations at $k = 400\text{N/m}^2$ which look much more irregular. The counterargument to this is that we see the same regular oscillations at $k = 2000\text{N/m}^2$, well above the inverted region (figure 13). It is also unclear what mechanism would create a resonance frequency in this particular region.

In the region $k > 1500\text{N/m}^2$, the relationship is yet again approximately linear, but this time more or less independent of γ. Above this region, it might flatten out and approach 100% as k approaches infinity.

In conclusion, our simulation showed that the relationship between k and bounce height has a similar shape regardless of damping, and that the relationship is approximately linear for small and large k. However, possible flaws in the model could cast doubt on these results.

Bounce height as a function of k and γ

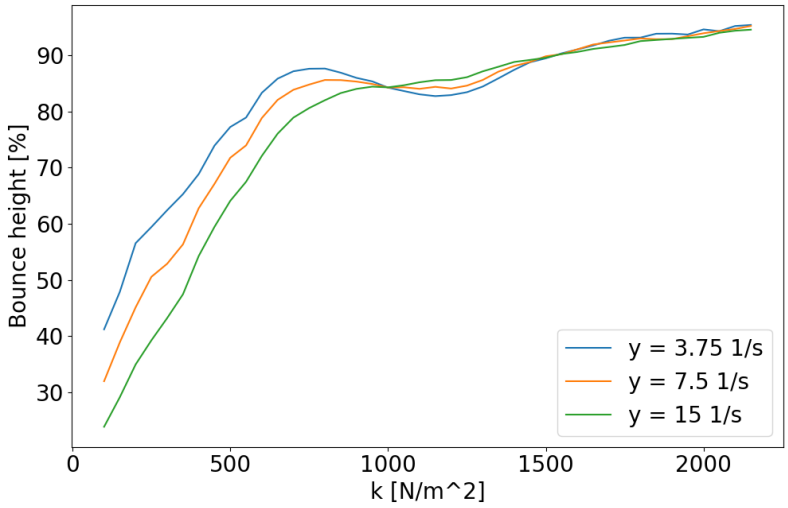

Figure 10: Bounce height (as % of initial height 10m) plotted as a function of k for 3 different values of γ.

Potential energy stored in springs, $k = 400\mathrm{N/m}^2$

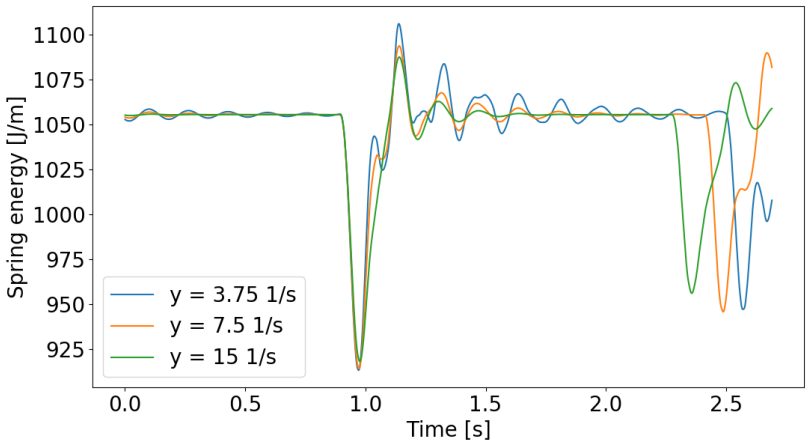

Figure 11:

Potential energy stored in springs, $k = 1250\mathrm{N/m}^2$

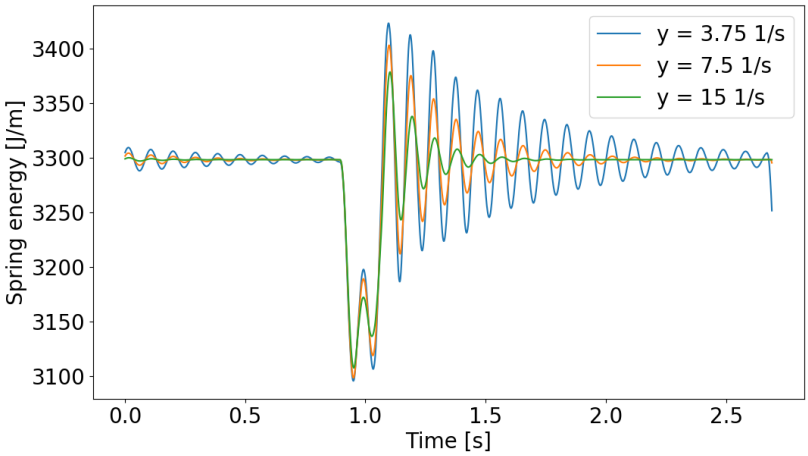

Figure 12:

Potential energy stored in springs, $k = 2000\mathrm{N/m}^2$

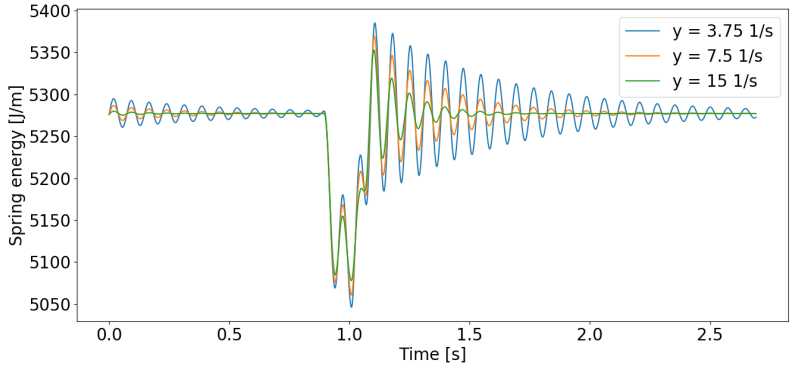

Figure 13: