The Simulation of Rattleback's Movements in Two Directions

Yuqian Zhang (yz6572@nyu.edu)
Yifei Zhu (yz6994@nyu.edu)

October 2022

Contents

1 Introduction

The rattleback (sometimes called a celt or cobblestone) is a semi-ellipsoidal solid that prefers to rotate in a certain direction. The amazing behavior is when the rattleback rotates in the other direction, it firstly finishes this rotation with oscillations and reverses spin direction, and finally stops as rotating in the preferred direction.

This topic was first discussed by G.T. Walker (1896), who showed that inertial asymmetry would cause a change in the structure of dynamic equations. In 1986, Bondi provided a detailed explanation starting from Walker's assumption and extended it with the conservation of energy [1]. Our project primarily follows Bondi's description but with some simplifications on the notations and calculations.

Our project aims to explain this counter-intuitive motion by analyzing the transfer of energy between the spin and oscillations, and to avoid the complicated calculation, we use energy conservation in the part of analyzing the reversal of the rattleback. We use the simulate this process and control the initial angular velocity and friction coefficient to observe the rotation under different situations.

2 Equations for the Physical Model

2.1 Assumptions and the Outline

In our physics model, we assume the rattleback has a smoothly curved lower surface when spinning on a smooth horizontal surface. Also, the contact point is relatively statistic to the ground so that the mechanical energy is conserved, but the contact points will move among the curved lower surface. The first part of our model analyzes how instabilities would occur in the oscillations.

2.2 Notations

The notations are as follows:

Symbol	Definition
M	mass of the rattleback
f	contact force
$\mathbf{v}$	velocity of the center of mass
$\boldsymbol{\omega}$	angular velocity of rattleback
$\boldsymbol{L}$	angular momentum of rattleback
$\boldsymbol{r}$	the position of the contact point with the ground
$\boldsymbol{u}$	unit normal vector to the rattleback's surface at the contact point
n	spin, with the equation $n = \boldsymbol{u} \cdot \boldsymbol{\omega}$

Table 1. Symbols and Definitions

2.3 Model explanations and equations

Firstly, we set up the coordinate for the rattleback with its center of mass as the origin. Set the short axis direction as the x-axis, and the long axis as the y-axis. We use vector $\boldsymbol{r}$ to express the location of the contact point: $\boldsymbol{r} = x\boldsymbol{u_x} + y\boldsymbol{u_y} + z\boldsymbol{u_z}$, where $\boldsymbol{u_x}$, $\boldsymbol{u_y}$, and $\boldsymbol{u_z}$ are unit vectors, and $\boldsymbol{u_z}$ has the downward direction which starts from the center of mass. Thus, under this condition, we have the lower surface of the rattleback [2]:

$$z = a\left[1 - \frac{1}{2}p(\frac{x}{a})^2 - q\frac{xy}{a^2} - \frac{1}{2}s(\frac{y}{a})^2\right] \tag{1}$$

where a means the distance from the center of mass to the lower surface along the z-axis. The goal is to make q very small and $-\frac{q}{p}$ is the tangent of the angel. In the program, we take a, p, q, s as integers to make the calculation easier. Since the mass is uniformly distributed, we set $q = 0$ as the default. Since the contact point is on the lower surface, we'll get the expression of $\boldsymbol{u}$:

$$\boldsymbol{u} = -\left(\frac{px + qy}{a}, \frac{qx + sy}{a}, 1\right) \tag{2}$$

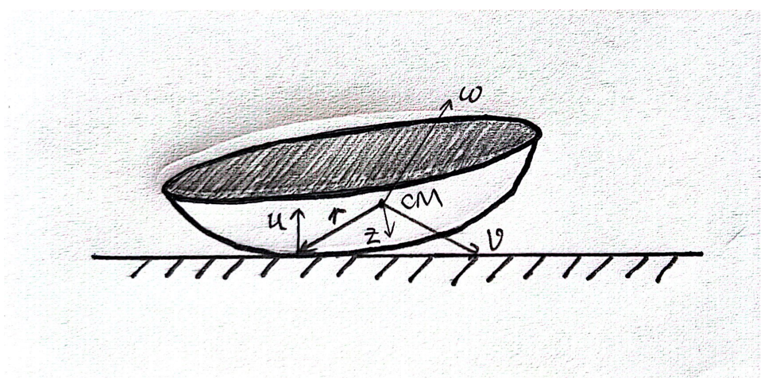

Fig. 1. rattleback

2.4 Before Reversal: from Spinning to Oscillations

Basically follow the article of Case and Jalal, we do the following calculation[2]. For the motion, initially, we let the rattleback rotate about the vertical axis with no oscillations. For the force and moment, we have the equations

$$M\frac{d\boldsymbol{v}}{dt} = \boldsymbol{f} - Mg\boldsymbol{u} \tag{3}$$

by Newton's second law, and

$$\frac{d\boldsymbol{h}}{dt} = \boldsymbol{r} \times \boldsymbol{f} \tag{4}$$

By the change of the system, we change from the rotating coordinate system changing with the rigid body to the ordinary coordinate system (rattelback-stable coordinate system) with the formula

$$\frac{d\boldsymbol{u}}{dt} = \dot{\boldsymbol{u}} + \omega \times \boldsymbol{u} \tag{5}$$

where $\boldsymbol{\omega}$ is angular velocity of the rattleback, and $\dot{\boldsymbol{u}}$ means the time derivative in the rotating coordinates. Since $\boldsymbol{u}$ is constant, the derivative of $\boldsymbol{u}$ is 0, and we bring it into the former equation to get the angular velocity $\boldsymbol{\omega} = \dot{\boldsymbol{u}} \times \boldsymbol{u} + n\boldsymbol{u}$ with $n = \boldsymbol{\omega} \cdot \boldsymbol{u}$. Then we take the equation (2) to this equation and get the angular velocity expression with x and y.

$$\boldsymbol{\omega} = \left[\frac{q\dot{x} + s\dot{y} - n(px + qy)}{a}, \frac{-p\dot{x} - q\dot{y} - n(qx + sy)}{a}, -n \right] \tag{6}$$

Then, we consider the velocity. The movements of points of the rattleback will be evaluated by the angular velocity relative to the center of mass. Thus, we can express the velocity of the center of mass in the body-fixed coordinate system is

$$\boldsymbol{v} = -\omega \times \boldsymbol{r} \tag{7}$$

We can take the formula (6) inside to get the formula of velocity with x and y.

Then, we take the derivative of velocity with respect to t to get the acceleration in each direction, so we have

$$\frac{dv_x}{dt} = p\ddot{x} + q\ddot{y} + 2nq\dot{x} - n(1 - 2s)\dot{y} \tag{8}$$

$$\frac{dv_y}{dt} = q\ddot{x} + s\ddot{y} + 2nq\dot{y} - n(1 - 2p)\dot{x} \tag{9}$$

$$\frac{dv_z}{dt} = 0 \tag{10}$$

As we discussed before, because of the change in the coordinate system, we have

$$\frac{d\boldsymbol{L}}{dt} = \dot{\boldsymbol{L}} + \omega \times \boldsymbol{L} \tag{11}$$

Also, $\frac{d\boldsymbol{L}}{dt}$ stands for the torque about the center of mass, so we'll get another equation about it:

$$\frac{d\boldsymbol{L}}{dt} = m\left[\boldsymbol{r} \times \frac{d\boldsymbol{v}}{dt} + g\boldsymbol{r} \times \boldsymbol{u} \right] \tag{12}$$

From equations (11) and (12), we can get the change of angular momentum in each direction, and combined with the formulae before, we can express the change with the equation that

$$(q\ddot{x} + s\ddot{y})\alpha - n(p\dot{x} + q\dot{y})(\alpha + \beta - \gamma) + n\dot{x} = \frac{g}{a}[-y(1 - s) + qx] \tag{13}$$

$$-(p\ddot{x} + q\ddot{y})\beta - n(q\dot{x} + s\dot{y})(\alpha + \beta - \gamma) + n\dot{y} = \frac{g}{a}[x(1-p) + qy] \tag{14}$$

Notice that the moments of inertia about the x, y, and z axes are

$$I_{11} + Ma^2 = \alpha Ma^2 \tag{15}$$

$$I_{22} + Ma^2 = \beta Ma^2 \tag{16}$$

$$I_{33} = \gamma Ma^2 \tag{17}$$

Here we consider I_{11} as I_x, I_{22} as I_y, I_{33} as I_z

We assume q is small, so ignore terms with q higher than one order, and we take $x = x_0 e^{i\Omega t}$ and $y = y_0 e^{i\Omega t}$ to do the calculation. The relation of frequencies is:

$$(\Omega^2 - \Omega_x{}^2)(\Omega^2 - \Omega_y{}^2) = i\frac{\Omega^3 nq(\alpha - \beta)}{\alpha\beta sp} \tag{18}$$

Then, we subsitute $\Omega = \Omega_x + \Delta_x$ and $\Omega = \Omega_y + \Delta_y$ to get

$$\begin{aligned}
(\Omega^2 - \Omega_x{}^2)(\Omega^2 - \Omega_y{}^2) &= [(\Omega_x + \Delta_x)^2 - \Omega_x{}^2][(\Omega_y + \Delta_y)^2 - \Omega_y{}^2] \\
&= (2\Omega_x\Delta_x + \Delta_x{}^2)(2\Omega_y\Delta_y + \Delta_y{}^2) \\
&= i\frac{(\Omega_x + \Delta_x)^3 nq(\alpha - \beta)}{\alpha\beta sp} \\
&= i\frac{(\Omega_y + \Delta_y)^3 nq(\alpha - \beta)}{\alpha\beta sp}
\end{aligned}$$

By solving this equation, we can get Δ_x and Δ_y for the oscillation condition of the asymmetry rattleback.

$$\Delta_x = i\frac{\Omega_x{}^2 nq(\alpha - \beta)}{2\alpha\beta sp(\Omega_x{}^2 - \Omega_y{}^2)} \tag{19}$$

and

$$\Delta_y = -i\frac{\Omega_y{}^2 nq(\alpha - \beta)}{2\alpha\beta sp(\Omega_x{}^2 - \Omega_y{}^2)} \tag{20}$$

By doing so, we can identify $\Omega_x{}^2$ and $\Omega_y{}^2$. The former will describe the oscillations about the x-axis, and the second term will tell the oscillation about the y-axis.

2.5 The Conservation of Energy for the Stop and Reversal

By far, we've answered how the rotation will change into oscillation, and we'd like to discuss what will happen after the rattleback oscillates acutely, the reversal. We use the conservation of energy to explain it. We can find ${\Omega_x}^2$ is larger than ${\Omega_y}^2$, so we mainly focus on the oscillation about the x-axis. Different from Ω_x, Ω_y is comparable to n so that it will take little energy, but oscillation about the x-axis will take up all the initial energy and make the system stop. The total initial energy is:

$$E_x = \frac{1}{2} I_z n^2 \tag{21}$$

where θ means the amplitude of motion, and the oscillation about the x-axis will grow and take all the energy:

$$E_x = \frac{1}{2} I_x (\Omega_x \theta_0)^2 \tag{22}$$

where Ω_x, the angular frequency, is much higher than n, but $\Omega_x \theta_0$ is comparable to n.

Then we come to the third phase. For an asymmetry rattleback, we can consider the cross-section as Fig.2. The force Mg will tend to rotate the rattleback to the right, so the center of mass will be accelerated to the right. Thus, there's a force at the contact point, which will make the rattleback rotate in a different direction.

3 Numerical Method

3.1 Determining the rattleback boundary and mesh properties

We built a polygon mesh model to realize the previous physical mechanisms. In our practice, we assigned equal mass to each mesh point inside the determined boundary and remove two small ellipsoids on the top surface.

Let's first focus on the boundary except the removed small paraboloids, which is a top plane and a bottom large paraboloid. According to the physical part, the formula for the paraboloid is

$$z = a\left[1 - \frac{1}{2} p (\frac{x}{a})^2 - q\frac{xy}{a^2} - \frac{1}{2} s (\frac{y}{a})^2\right] \tag{23}$$

Since the direction of z in the physical model and our coordinate system is in the opposite direction, we should consider this z here as $-z$, then this is equivalently

$$z = \frac{p}{2a} x^2 + \frac{q}{a^2} xy + \frac{s}{2a} y^2 - a \tag{24}$$

We set the parameters: $a = 4, p = 3, q = 0, s = 0.15$ and the top plane is fixed at $z = 2$. So the equation for x and y is

$$20x^2 + y^2 = 320 \tag{25}$$

This tells us the x and y mesh ranges are $[-4, 4]$ and $[-8\sqrt{5}, 8\sqrt{5}]$, and the ratio is $y^2/x^2 = \frac{p}{s} = 20$. These ranges are valid since the top surface takes up the most x and y values in our model. As for the z range, it is easy to see that z takes its minimum of $-a$ at $(x, y) = (0, 0)$. So z range is $[-a, 2] = [-4, 2]$.

Next, we go on to determine the mesh lengths. To simulate the uniform material, we should maintain an even mass distribution across the rattleback. That is to say, it is a must that we keep a uniform mesh length in each of x, y, and z dimensions. According to our previous calculation of the mesh ranges, the range in y direction is significantly larger. This reminds us to make the y-mesh length also larger.

Another important thing in designing uniform mesh length is that we should avoid "odd points" in the model. An odd point is a mesh point whose coordinates are within the rattleback boundary, but significant different from other points around it. One case of an odd point would be the lowest point on the bottom paraboloid. If we allow both the x and y mesh to take 0 as a mesh point, and take a regular z mesh length such as 0.5, then only one point $(0, 0, -4)$ could be reaching the lowest z-value of -4, and all other points closely around it has z-value of -3.5. In this case, our model is essentially an approximated paraboloid with one particular point pointing out downwards. This might cause unexpected motion of the model. Similar situation may also occur to the leftmost, rightmost points. Our way to solve this is to make an odd number of intervals so that the mesh interval in the middle is symmetric about the origin. In this way no point could lie on the horizontal origin (i.e. has $(0, 0)$ in x and y coordinates). Our final mesh range is shown in the following table.

	lowest end	highest end	mesh length	mesh interval count
x	-3.9	3.9	0.52	15
y	-17.75	17.75	1.42	25
z	-4	2	0.2	30

3.2 Adjusting mass distribution by removing two small paraboloids

We set the total weight of the rattleback be 500 grams. We want this mass to be distributed evenly across the rattleback. To achieve this, not only do we need to keep uniform mesh lengthes, but in the final step we are to divide the total weight by the total count of mesh points so that each point is weighed equally. Next, we want the mass to be asymmetric about x and y axis by removing two small semi-paraboloids from the top plane, on different sides of the long axis.

The size of two small semi-paraboloids is set to be $\frac{3}{8}$ of the whole rattleback. After that, we shall move two semi-paraboloids to the proper locations by substituting $x = x + m$ and $y = y + n$. Since we want the small semi-paraboloids also to be symmetric with out mesh grid, the constants m and n should be several times of our x and y mesh lengths. In the program we specify $m = 2.0 * \text{x-mesh} = 2.0 * 0.52 = 1.04$ and $n = 6.5 * \text{y-mesh} = 6.5 * 1.42 = 9.23$.

In the following figures we show different views of the rattleback. Here we only draw out the points on the outer surface, but our actual model also consists of all the points in the middle. There are a total of 4256 points taken here, and 602 of them on the surface are drawn out. Red points are on the top plane and blue ones are on the bottom paraboloid.

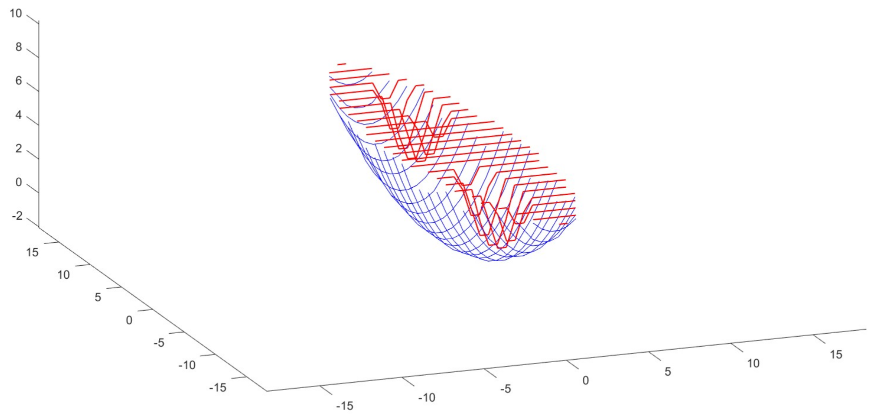

Fig. 2. Oblique view

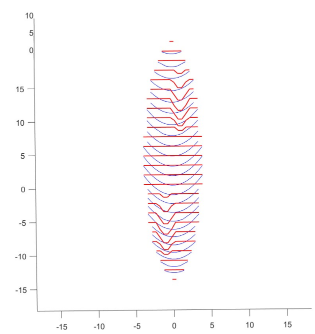

Fig. 3. Above view

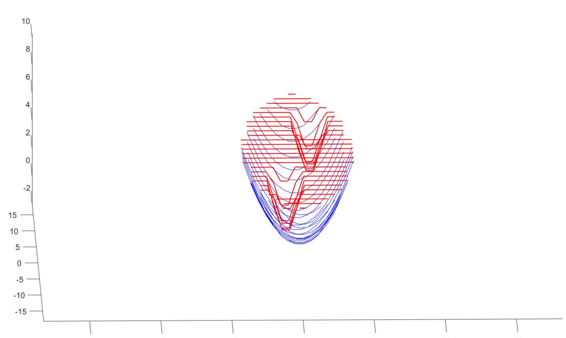

Fig. 4. Front view

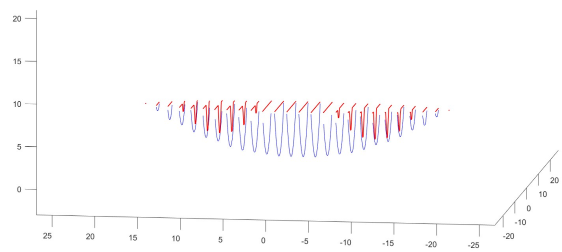

Fig. 5. Side view

3.3 Simulating the force

When the rattleback starts moving, we consider four components of the force: elastic force, damping, gravity, and friction. Initially, there's only gravity acting on the rattleback with the expression $-Mg$. The gravity is constant in the motion since no change to the mass. To be specific, each mesh point is assigned the gravity of its own mass. The other three forces are acted by the ground and only to the points below the ground. Elastic force, $-kx$ is proportional to the distance below the ground x by a spring coefficient. Damping and the velocity of the rattleback have a positive correlation by a damping constant. We only add damping to a point when it is below the ground. The total normal force of the ground is the sum of the two and we require it must be positive since the ground can't be pulling an object into itself. Lastly, the friction is proportional to the total normal force by a kinetic friction constant. The next section will introduce how we simulate the total force and the movement.

3.4 Updating the variables

We use a regular first order Euler method to simulate the forces between ground and rattleback. The overall framework is similar to Professor Charles Peskin's lecture notes on rigid body dynamics[5]. In our case, we consider the rattleback, a rigid body, as a list of points whose positions are expressed by the position of center of mass $\vec{X_{cm}}$ and positions of points relative to the center of mass $\vec{\tilde{X}}$, by the expression $\vec{X} = \vec{X_{cm}} + \vec{\tilde{X}}$.

We update $\vec{X_{cm}}$ and $\vec{\tilde{X}}$ in different ways: obtaining the $\vec{X_{cm}}$ by

$$\frac{d\vec{X_{cm}}}{dt} = \vec{U_{cm}} \tag{26}$$

$$\frac{d\vec{U_{cm}}}{dt} = \frac{\vec{F_{total}}}{M} \tag{27}$$

where $\vec{F_{total}}$ is the total external force (we consider the friction, the gravity, and the support force from the ground).

For calculating $\vec{\tilde{X}}$, we first determine the angular velocity from the angular momentum L with moment of inertia

$$I\Omega = L \tag{28}$$

Then we consider how to describe the interaction with the ground. There are two crucial vectors: the total force $\vec{F_{total}}$ and torque $\vec{\tau}$, both caused by the ground to the body but the former acts on the center of mass and the torque works on the momentum.

Similar to the model of the motion of wheel, we consider each point below the surface with the force $\vec{F_k}$ so the total force can be expressed as

$$\vec{F_{total}} = \sum^{k} \vec{F_k} \tag{29}$$

so the torque is

$$\vec{\tau} = \sum^{k} (\vec{\tilde{X}_k} \times \vec{F_k}) \tag{30}$$

To calculate the individual force $\vec{F_k}$, we follow the notes that [3]

$$\vec{F_k} = S \frac{-H(\vec{X})}{\|\nabla H(\vec{X})\|} \frac{\nabla H(\vec{X})}{\|\nabla H(\vec{X})\|} - \mu S \frac{-H(\vec{X})}{\|\nabla H(\vec{X})\|} \frac{\vec{U^{tan}}}{\|\vec{U^{tan}}\|} \tag{31}$$

where $H(\vec{X})$ means the ground, here we use $z = 0$ to stand for the ground, with $z > 0$ above the ground and $z < 0$ below the ground (we change the direction in the beginning). And μ is the coefficient of sliding friction. Also, similar to $\vec{X_k}$, $\vec{U_k}$ is

$$\vec{U_k} = \vec{U_{cm}} + \vec{\tilde{U}_k} \tag{32}$$

where $\vec{U_k} = \Omega \times \vec{\tilde{X}_k}$

4 Validation

4.1 Structure

The rattleback's structure should fit all four following requirements [4]:

1. The lower surface of the rattleback should be the paraboloid, and one direction's curvature radius should be larger than the other (the shorter axis).
2. Each axis in the coordinate of inertia has a certain degree from the coordinate of the body, which means that the mass distribution is asymmetry with the coordinate of the body.
3. There's friction between the surface and the lower surface of the rattleback so there could be rolling friction.

Here's the calculation for validating the conditions of the structure:

1. For our system, we need to show that the curvature radius of the x-axis ρ_x is smaller than that for the y-axis ρ_y. Based on the formula

$$R = \frac{(1 + (\frac{dy}{dx})^2)^{\frac{3}{2}}}{|\frac{d^2y}{dx^2}|} \tag{33}$$

   with x as the parameter.
   The result for $\rho_x = \frac{2\sqrt{5}}{5}$ and $\rho_y = 80$, which obviously shows that $\rho_x < \rho_y$.

2. The angular is not zero by removing two semi-paraboloids.

3. The coefficient of friction is $\rho = 0.3$ and 0.5 based on our program.

4.2 Energy

When there's no friction between the surface and the rattleback, the energy should be consistent all the time. Here's the picture of time and total energy (which is given by the initial energy):

$$E_{rotational} = \frac{1}{2} I \omega^2 \tag{34}$$

where $I = (1.0e+04) \cdot \begin{bmatrix} 2.9781 & 0.0648 & -0.0000 \\ 0.0648 & 0.2545 & -0.0000 \\ -0.0000 & -0.0000 & 3.0101 \end{bmatrix}$ and $\omega = 10$ rad/s, so the total energy provided by rotational kinetic energy is close to 3×10^6. The following picture shows the energy change with time.

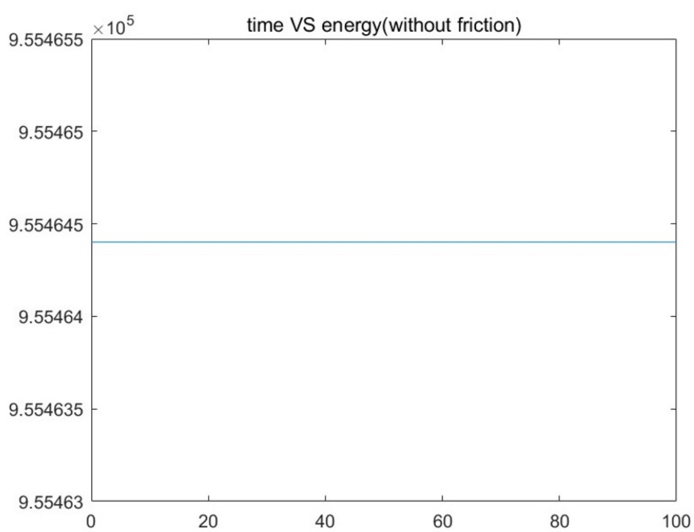

Fig. 6. Time and energy without the friction

And under the condition of friction, the line decreases.

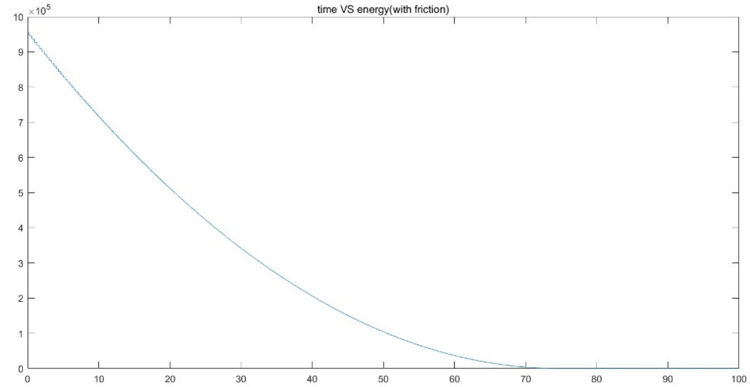

Fig. 7. Time and energy with the friction: preferred direction

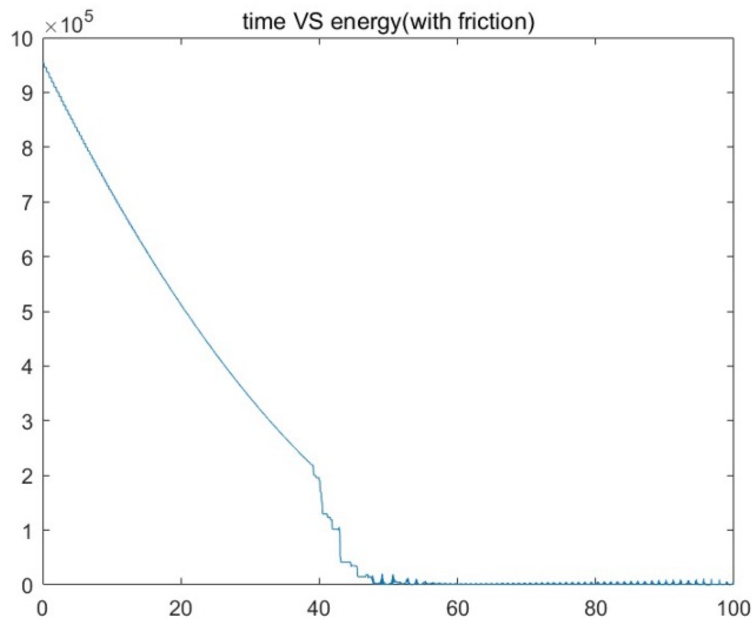

Fig. 8. Time and energy with the friction: reverse direction

These two pictures prove that our model will have different results for the preferred rotating direction and the other direction. The regular direction follows a smooth quadratic decrease, and the reverse one drops quickly after about 40 seconds. The graph in Fig. 7 is smoother than that in Fig. 8. In the second half of the graph in the reverse direction, since our program will end with an unusual rolling, so the graph will be zig-zag. However, it still follows the physical rule since it will get close to 0 and finally stops on the ground.

4.3 Movement before removing semi-paraboloid

Before discussing the movements in both directions, we first test the model without two semi-paraboloids shown below.

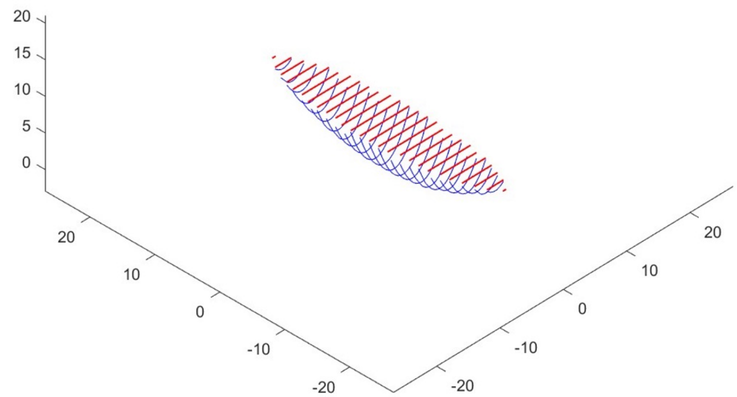

Fig. 9. rattleback without two semi-paraboloids

This means there's no degree between the geometric axis of the rattleback and the inertia axis. However, we still keep the friction in our model to make it stop. Under these conditions, one point's movement in x, y directions will be periodic due to the oscillation.

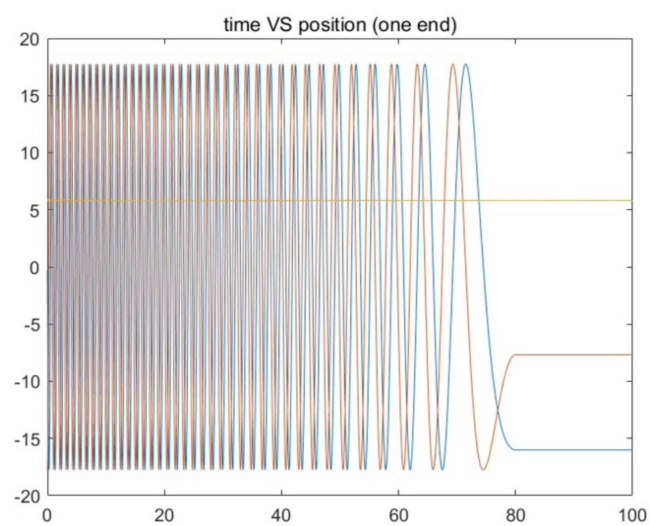

Fig. 10. Time and position

This picture tracking on the point on one end (to make the graph of movement more obvious) proves our predictions. The point will oscillate among the x and y directions. There's no oscillation along the z direction since the mass is uniformly distributed and the geometric axis and inertia axis lie in the same line.

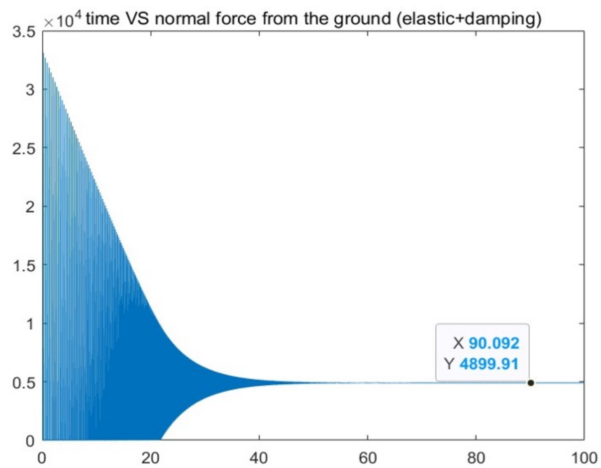

Fig. 11. Time VS normal force (elastic and damping)

4.4 Movement after removing semi-paraboloid

We recover the standard model with two removed semi-paraboloids on the top. By the physical modeling part, we've shown that there'll be rotation and oscillation in each x, y, and z axis direction.

In Fig. 12, the blue and red lines stand for the point's x and y movement, and the yellow line means the track of z-axis position. In the beginning, all these lines describe the harmonic motion, and when time reaches 40, the lines' fluctuation starts to decay and finally reaches a relatively stable state. This means that, with friction, the rattleback will stop in the end.

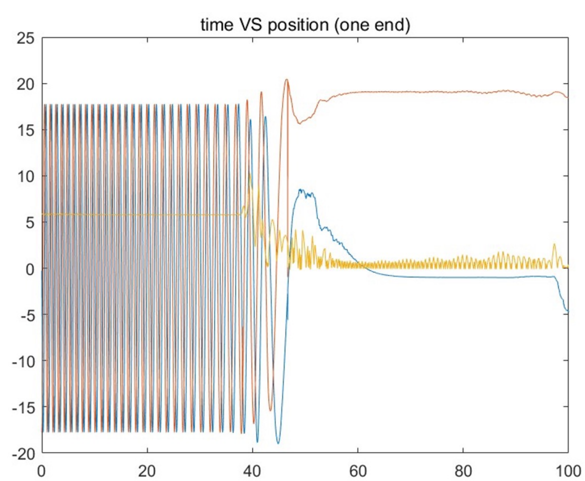

Fig. 12. Time and position with semi-paraboloids

Figure 13 shows that the angular velocity of rattleback decreases (by absolute value, the minus sign here denotes orientation) linearly from 0 to 40 seconds. This is due to the friction effect. It then drops rapidly and stays around 0, which corresponds to the turnover.

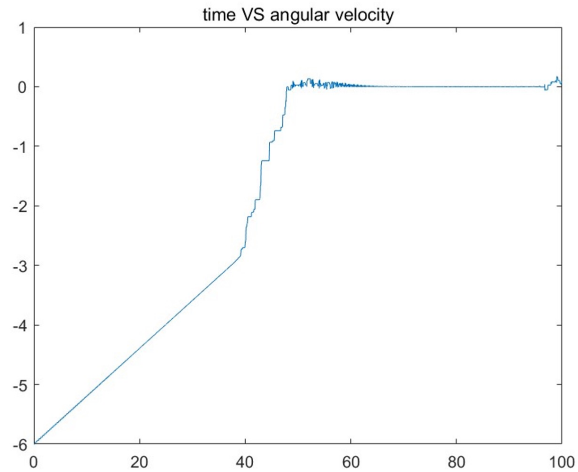

Fig. 13. Time and angular velocity with semi-paraboloids

As we zoom in the previous figure, we can see that the curve actually exhibits a step-like behavior at the very beginning. Then both the horizontal and vertical step-size are decreasing and the function is more smooth. This is also validated by the damping force we added to the model.

When we think about the theoretical explanation of this graph, we should consider the damping force, which has the opposite direction from the one of velocity as we discussed before. Since the damping will prohibit movements in the up-and-down direction, the step size in the x and y directions will be smaller as time increases. Intuitively, the model will first "jump" more highly above the ground and reach a deeper position under the ground when it falls. But only when it is below or on the ground the rattleback's movement will be affected by the friction which will result in a change of angular velocity. Thus, in the beginning, the function has an obvious step-like shape. But as time *t* becomes larger, the rattleback will not jump as high as its height in the previous steps but will have a higher frequency. And finally, the model will be stable in the z direction, which means that it will remain at a constant height relative to the ground.

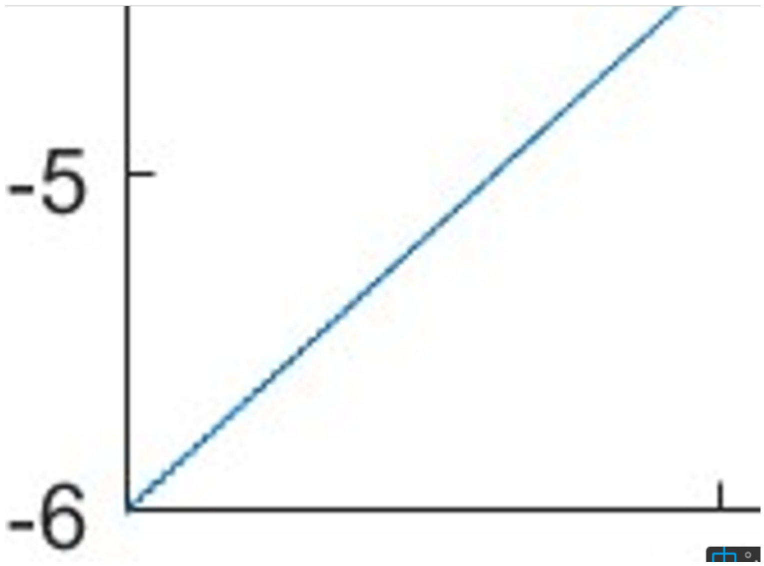

Fig. 14. Zoom-in figure for time and angular velocity

The following figure shows how the position of lowest point changes. We can see that the z-range (yellow curve in the graph) is gradually getting smaller and approaching 0 stably from the beginning, which meets our expectation due to damping. The "wavelength" of red curve and blue curve is increasing from 0 to 40 seconds. That is, the periodic curve is less dense. This is behavior is also physical, due to friction.

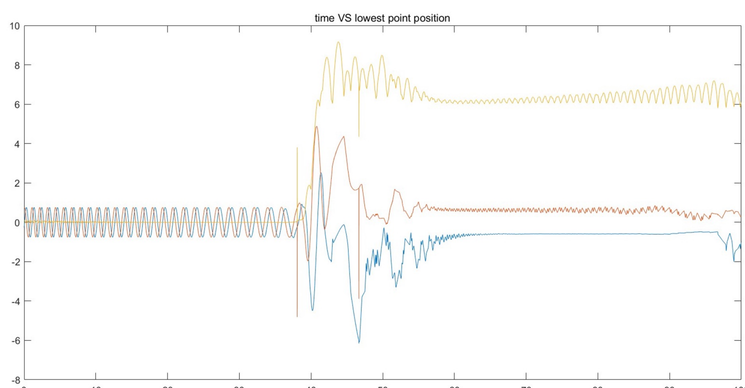

Fig. 15. Time and low point position

In this figure the vertical axis is the x,y,z component of the total force on the lowest point. The yellow line, as the z component, is always positive and decreases almost linearly to about $\frac{1}{8}$ of the gravity of the whole rattleback (there are a total of 8 points which are all lowest points. $500 * 9.8/8 = 612.5$, the yellow line is approaching this value after 30 seconds).

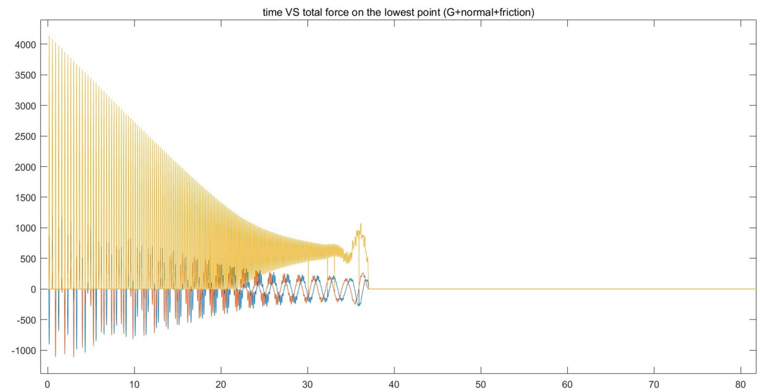

Fig. 16. Time and total force on the lowest point

5 Discussion

The video of our final simulation is here. Here the regular orientation (clockwise when viewed from above) acts physically and gradually stops with friction; the reverse orientation (counter-clockwise when viewed from above) is a little unphysical since it actually flips over but it at least realizes the transition from rotation to some other oscillation. Maybe due to some parameter issues, such oscillation is too strong that the model is flipped.

We tried to use two different friction coefficients here, 0.3 and 0.5, to find the effect on the dynamics. Both cases are run in the reverse orientation, the "odd" one. The resulting video is here. In this video, we can see that with a higher friction coefficient, the model will stop a little more quickly. When considering the energy consumption, with the same initial condition, the higher friction coefficient will cause greater friction, which will prohibit the rattleback's movement and consume more kinetic energy. Also, in our previous tests for dozens of times, we find that the larger the friction coefficient we set, the more regular our model will be: it will rotate stably and stop on the ground. This is also shown in the video as the 0.5 case seems more stable than the 0.3 case even after they both flip over.

6 Summary and Conclusions

In our program, we simulate the properties of rattleback, which fits the physical structure. We also make the rattleback rotate with a certain initial angular velocity in the preferred direction and the non-preferred one. The videos and graphs prove the difference between movements in the two directions, which is illustrated by theoretical equations in the physical part. The result shows that in the preferred, the rattleback's movement is relatively stable and normal, while in the other direction, the movement will combine with more oscillations and finally turns over, which is a relatively unstable state.

7 References

References

[1] Garcia, A., and M. Hubbard. "Spin Reversal of the Rattleback: Theory and Experiment." Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, vol. 418, no. 1854, 1988, pp. 165–97. JSTOR, http://www.jstor.org/stable/2398321. Accessed 30 Oct. 2022.

[2] Case, William, and Sahar Jalal. "The Rattleback Revisited." American Journal of Physics, vol. 82, no. 7, 2014, pp. 654–658., https://doi.org/10.1119/1.4869286.

[3] Peskin, Charles S. "Interaction of Dynamic Structures with the Ground: Non-Penetration and Sliding Frictional Forces." MODELING AND SIMULATION IN SCIENCE, ENGINEERING, AND ECONOMICS, https://www.math.nyu.edu/ peskin/modsim lecture notes/interaction with ground.pdf.

[4] Ding, Zhanao. "Mechanics Principles of the Rattleback." China Academic Journal Electronic Publishing House, vol. 15, 1993, pp. 74–77.

[5] Peskin, Charles S. "Dynamics of Rigid Bodies" MODELING AND SIMULATION IN SCIENCE, ENGINEERING, AND ECONOMICS, https://www.math.nyu.edu/ peskin/modsim lecture notes/rigid body motion.pdf

8 Appendix

Here's the code for simulating the rattleback structure.

```
function [elecnt,surfacecnt,lmax,toplmax,X,jj,kk,M] = ...
rattleback(shortlen, ratio, asymratio, shfdsft,
lofdsft, meshes, mass, paras)

toplevel = 2;
RHS = ratio * shortlen^2;
longlen = sqrt(RHS);
xmesh = meshes(1);
ymesh = meshes(2);
zmesh = meshes(3);
xrange = -shortlen+0.1:xmesh:shortlen-0.1;
yrange = ceil(-longlen)-0.75:ymesh:floor(longlen)+0.75;
zrange = 2:-1 * zmesh:-paras(1);
a = paras(1);
p = paras(2);
s = paras(4);
strx = append(num2str(p), '/2/', num2str(a), '*');
stry = append(num2str(s), '/2/', num2str(a), '*');
espd_string = append(strx, 'x^2+',stry, 'y^2-', num2str(a));
%disp(espd_string)
function_string = ['@(x,y)' espd_string];
bottom_espd = str2func(function_string);

if asymratio ~= -1
    %small_shlen = shortlen * asymratio
    %small_lolen = longlen  * asymratio
    shsft = num2str(shfdsft * xmesh);
    losft = num2str(lofdsft * ymesh);
    small_strx = append(num2str(p), '/2/', num2str(a),
    '/', num2str(asymratio^2), '*');
    small_stry = append(num2str(s), '/2/', num2str(a),
    '/', num2str(asymratio^2), '*');
    small_espd_string1=append(small_strx,'(x',shsft,')^2+',
    small_stry,'(y-',losft,')^2-');
    small_espd_string1=append(small_espd_string1,
    num2str(a-0.1), '*', num2str(asymratio));
    small_espd_string2=append(small_strx,'(x+',shsft,')^2+',
    small_stry,'(y+',losft,')^2-');
    small_espd_string2=append(small_espd_string2, num2str(a-0.1),
    '*', num2str(asymratio));
    function_string1 = ['@(x,y)' small_espd_string1];
```

```
    function_string2 = ['@(x,y)' small_espd_string2];
    small_espd1 = str2func(function_string1);
    small_espd2 = str2func(function_string2);
end

% the additional 4th dimension denotes whether element is on the surface
X = zeros(length(xrange)*length(yrange)*length(zrange), 4);
elecnt = 0;
surfacecnt = 0;
for j = 1:length(yrange)
    for i = 1:length(xrange)
        x = xrange(i);
        y = yrange(j);
        zbound = bottom_espd(x,y);
        if zbound > 2
            continue
        end
        %fprintf('x = %f, y = %f, zbound = %f\n', x,y,zbound);
        real_bound = 100;
        if asymratio ~= -1
            zbound1 = small_espd1(x,y);
            zbound2 = small_espd2(x,y);
            if zbound1 > 2 && zbound2 <= 2
                real_bound = zbound2;
            elseif zbound1 <= 2 && zbound2 > 2
                real_bound = zbound1;
            end
        end
        for k = 1:length(zrange)
            outsidesmall = 1;
            z = zrange(k);
            if z < zbound
                break;
            end
            if real_bound ~= 100 && z >= real_bound
                outsidesmall = 0;
            end
            if outsidesmall == 1
                elecnt=elecnt+1;
                X(elecnt, 1) = x;
                X(elecnt, 2) = y;
                X(elecnt, 3) = z;
                %fprintf('elecnt = %f, x = %f, y = %f,
                zbound = %f,z= %f\n', elecnt,y,zbound,z);
                if z == 2.0
```

```
                    X(elecnt, 4) = 1; % this element is on the top
                    surfacecnt = surfacecnt+1;
                elseif abs(X(elecnt-1, 1) - X(elecnt ,1) + xmesh) < 0.0001
                    X(elecnt, 4) = 1; % this element is on the top,
                    inside the small dent
                    surfacecnt = surfacecnt+1;
                elseif  z-zbound < zmesh
                    X(elecnt, 4) = 2;
                    % this element is on the bottom
                    surfacecnt = surfacecnt+1;

                end
            end
        end
    end
end

%for k = 1:length(zrange)
%    z = zrange(k);
%    for i = 1:length(xrange)
%
%        x = xrange(i);
%        if ()
%    end
%end
fprintf("elecnt = %d\n", elecnt);
fprintf("surface count = %d\n", surfacecnt);

X = X(1:elecnt, :);
M = (mass/elecnt)*ones(elecnt,1);
lmax = 2 * surfacecnt;
jj=zeros(lmax,1);kk=zeros(lmax,1);
l = 1;
this1 = 0; last1 = 0; this2 = 0; last2 = 0;
for i = 1: elecnt
    if X(i, 4) == 1
        this1 = i;
        if last1 == 0
            last1 = this1;
            continue;
        end
        %fprintf("last: (%d, %d) this: (%d, %d)\n", X(last1, 1),
        X(last1, 2), X(this1, 1), X(this1, 2));
        if X(last1, 1) - X(this1, 1) > 0
            last1 = this1;
```

```
            continue;
        end
        jj(l) = this1;
        kk(l) = last1;
        last1 = this1;
        l = l+1;
    end
end
toplmax = l-1;
for i = 1:elecnt
    if X(i,4) == 2
        this2 = i;
        if last2 == 0
            last2 = this2;
            continue;
        end
        if X(last2, 1) - X(this2, 1) > 0
            last2 = this2;
            continue;
        end
        jj(l) = this2;
        kk(l) = last2;
        last2 = this2;
        l = l+1;
    end
end
jj = jj(1:l-1, 1);
kk = kk(1:l-1, 1);
lmax = l-1;
fprintf("total link = %d\n", lmax);
fprintf("top link = %d\n", toplmax);

% n = number of points on the rim
% M_rim = total mass of the rim
% M_axle = total mass of the axle
% S_rim = stiffness of each rim link
% D_rim = damping constant of each rim link
% S_spoke = stiffnes of each spoke
% D_spoke = damping constant of each spoke
% S_axle = stiffness of the axle
% D_axle = damping constant of the axle

%outputs:
% X(k,:)      = coordinates of point k
% jj(l),kk(l) = indices of points connected by link l
```

```
% M(k)         = mass of element k
% elecnt       = total number of elements
% surfacecnt   = total number of elements on the top or bottom
% lmax         = total number of links
```