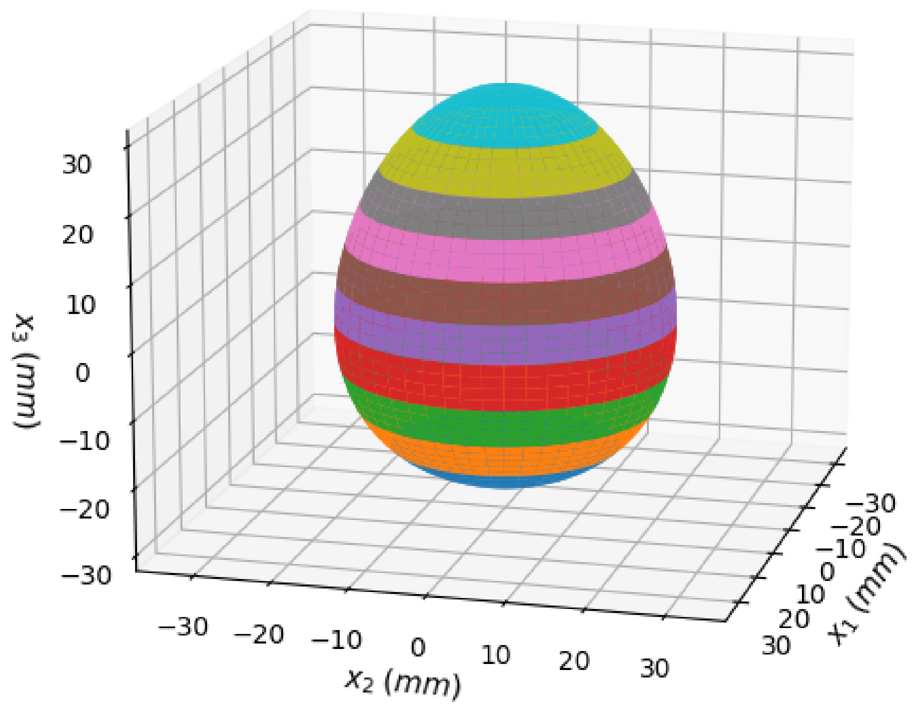
Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

The spinning egg: free and forced dynamics

Author: Stijn Wolters



Supervisors:

Prof. Dr. H.E. (Huib) de Swart
Utrecht University

Prof. Dr. L.R.M. (Leo) Maas
Utrecht University

Bachelor Thesis | July 3, 2021

Abstract

In this thesis an understanding of the motion of spinning eggs is developed. First the movement of a freely rotating egg is studied and after that the interesting forced rise of a spinning egg is investigated.

A mathematical description of the egg and its rotation is established to model both situations. From those the equations of motion are derived separately and multiple simulations are performed with different initial conditions. For the freely rotating egg the numerically solved model accurately describes the expected motion. In the case of an egg, approximated as ellipsoid, constrained to move on a table where friction plays its role, the numerical solution describes rolling and/or slipping effects realistically. The full rising effect, however, does not follow out of the model because of limits in its complexity.

Despite the fact that the full rising effect does not follow from our model, this thesis develops an interpretation about the behaviour of the rolling and slipping ellipsoid with the derivation and numerical solution of the equations governing its motion.

Contents

Acknowledgments

I would like to express my sincere gratitude to my supervisors Huib de Swart and Leo Maas. I very much appreciated their continuous enthusiasm throughout the entire project. They were both very keen in helping me trough every step of the research and were always positively supporting me.

I would also like to thank my parents for helping me through all kind of challenges over the course of my studies. Finally, I really appreciate the help of my brothers with proofreading and finalizing my thesis.

1 Introduction

There are interesting things going on when spinning objects in a three dimensional space. Some of these things can be perceived as strange at first glance, but they can be perfectly described and explained by physics and its theories. Currently, a vast amount of information about the behaviour of rotating objects can be found on the internet, including illustrating videos. An example of this is the phenomena of the unstable rotation along the intermediate principal axis of an object which is explained by the intermediate-axis theorem, more commonly known as the tennis racket theorem [Ashbaugh et al., 1991]. This theorem describes the fascinating behaviour of the 'Dancing T-handle in zero g'[i]. Other phenomena where spin is involved are, among others: the balancing spinning top[ii], the wobble of an American football or frisbee[iii], the balance of a bicycle wheel[iv]. These phenomena follow from an important law in physics: the conservation of angular momentum. Also a well known object that makes use of this conservation law and which has lots of applications in different fields is the gyroscope[v]. Now that some fascinating phenomena about spinning objects are introduced let us come to the subject of this thesis.

In this thesis the spinning of a very specific object is studied. As the title already suggests this object is an egg. There are two different aspects of a spinning egg that will be investigated in this thesis. The first is about the dynamics of a freely rotating egg. Imagine someone standing on the ground throwing a boiled egg freely into the air. Fun fact: there are actually world championships of who can throw the egg the farthest while another person aims at catching it without breaking the shell[vi]. Of course the egg will rise and after a while will fall down to the earth, but apart from that the egg will also rotate. How it rotates depends on the initial conditions, thus the motion the thrower applies to the egg. This leads us to the research question of the first problem:

How does an egg behave if it is thrown freely into the air? Can we solve the equation governing its motion and simulate its translation and rotation, which follow from the initial conditions?

This first problem is a stepping stone towards the second one. The second problem concerns itself with the remarkable forced rise of a spinning egg. It is described as follows: suppose one has a boiled chicken egg and spins it rapidly on a table with its symmetry-axis horizontal, the symmetry-axis from the egg will then rise from horizontal to vertical. This phenomenon is generally experienced as counter intuitive, which makes it an interesting phenomenon to conduct research on. A slow-motion video of this phenomena with an Easter egg is added in the footnote, serving as an example to aid understanding. [vii]. This phenomenon is mentioned by Moffatt and Shimomura [2002], who also give a short solution to the problem. In this thesis we aim to develop a better understanding of this phenomenon, making it clearer by actually solving the equations of motion numerically and providing additional insight by making animations that follows from this numerical solution. This leads us to the research question of the second problem:

[i]Dancing T-handle in zero g: `https://www.youtube.com/watch?v=1n-HMSCDYtM`
[ii]Spinning top: `https://www.youtube.com/watch?v=Mekf0oycidk`
[iii]Wobbling football: `https://www.youtube.com/watch?v=sgR2cmo2uB4`
[iv]Balancing wheel: `https://www.youtube.com/watch?v=8H98BgRzp0M`
[v]Gyroscope: `https://www.youtube.com/watch?v=cquvA_IpEsA`
[vi]Egg throwing championships: `https://www.youtube.com/watch?v=C-quyjYmGIY`
[vii]Rise of the egg: `https://www.youtube.com/watch?v=NnTHvTv4374`

Can we understand the physics of the rise of the spinning egg on a table and describe and simulate its movement by solving the derived equation of motion numerically?

The remainder of this thesis is organized as follows. First, the mathematical description of the egg, the calculation of its moments of inertia and the description of the orientation of the egg relative to a fixed coordinate frame are explained in section 2. Secondly, in section 3 the differential equation governing the motion of the freely rotating egg is derived and numerically solved. In this same section the results for this first problem are obtained and discussed. Thirdly, in section 4 the same is done for the motion of an egg constrained to roll and/or slide on a table, from which, with the right initial conditions, the rising effect should occur. In this second problem the shape of the egg is actually changed to that of an ellipsoid, which is explained in this section as well. Fourthly a conclusion is drawn for both problems in 5. Lastly, some of the Python code that I wrote and used is attached in the appendices.

2 Preliminaries

2.1 Size and mass egg

There is need for a mathematical description of a three-dimensional egg. In this thesis, we make use of the mathematical curve that is studied by and named after Fritz Hügelschäffer. The parametrization and construction of the Hügelschäffer egg are described on the website mentioned as a footnote [viii]. Some additions to this parametrization are made to get to our three dimensional mathematical egg. First, the parametrization is made, as needed, three-dimensional around the egg's symmetry-axis. Second, the orthogonal axes are named 1 (x_1), 2 (x_2) and 3 (x_3), where the 3-axis is the symmetry-axis and due to axis-symmetry, the 1-axis can be freely chosen. Third, the parametrization is shifted along the symmetry-axis by $x_{3,\text{shift}}$ such that the origin is at the centre of mass of the egg, where the calculation of the centre of mass is shown later in this chapter. The parametrization, in terms of Cartesian coordinates, reads

$$\frac{((x_3 + x_{3,\text{shift}}) - d)^2}{a^2} + \frac{x_2^2 + x_1^2}{b^2}\left(1 + \frac{2d(x_3 + x_{3,\text{shift}}) - d^2}{a^2}\right) = 1 \tag{1}$$

and

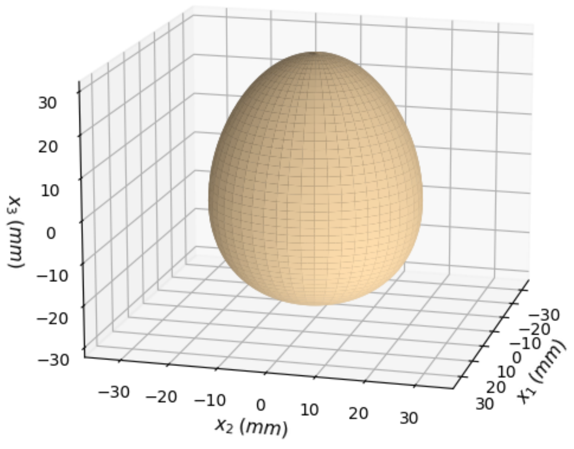
$$x_1 = b\sin(\alpha)\cos(\beta) \tag{2a}$$

$$x_2 = b\sin(\alpha)\sin(\beta) \tag{2b}$$

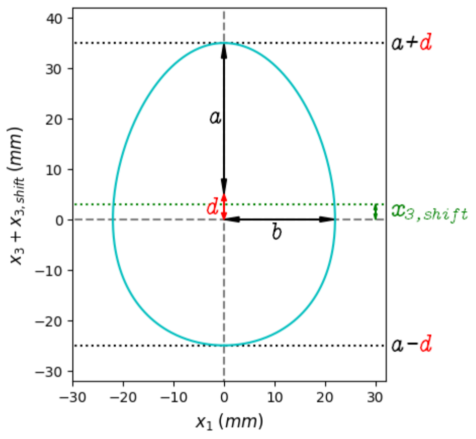
$$x_3 = \left(\sqrt{a^2 - d^2\sin^2(\alpha)} + d\cos(\alpha)\right)\cos(\alpha) - x_{3,\text{shift}} \tag{2c}$$

$$\text{where} \quad \alpha = [0, \pi\rangle \ , \ \beta = [0, 2\pi\rangle$$

If $d = 0$ in the equations, the parameterization is the same as that of an ellipsoid with a the semi-major axis and b the semi-minor axis and x_3, shift $= 0$ due to symmetry. So d ensures that the 'ellipsoid' is no longer symmetrical in the 1 3-plane and therefore gets its egg shape. Let us for example consider a chicken egg, which is shown in Figure 1a with the value of the parameters mentioned in the caption and illustrated in Figure 1b.



(a)



(b)

Figure 1: Hügelschäffer egg parameterization, with $a = 30$ mm, $b = 22$ mm and $d = 5$ mm

[viii]Hügelschäffer egg: `https://mathcurve.com/courbes2d.gb/oeuf/oeuf.shtml`

The volume of this egg is found by calculating the volume integral

$$V = \iiint_V dV = \int_{-a+d}^{a+d} \int_{-\left(\frac{b^2(a^2-(x_3-d)^2)}{a^2-d^2+2dx_3}\right)^{\frac{1}{2}}}^{\left(\frac{b^2(a^2-(x_3-d)^2)}{a^2-d^2+2dx_3}\right)^{\frac{1}{2}}} \int_{-\left(\frac{b^2(a^2-(x_3-d)^2)}{a^2-d^2+2dx_3}-x_2^2\right)^{\frac{1}{2}}}^{\left(\frac{b^2(a^2-(x_3-d)^2)}{a^2-d^2+2dx_3}-x_2^2\right)^{\frac{1}{2}}} dx_1\, dx_2\, dx_3 \tag{3}$$

Although this integral is analytically solvable, its expression is quite long. Therefore, it is chosen to leave it in its original form as shown above. The centre of mass of any rigid body is found by an integration over the volume as [Fowless and Cassiday, 2005]:

$$x_{3,cm} = \frac{\iiint_V \rho\, x_3\, dV}{\iiint_V \rho\, dV} = \frac{\iiint_V \, x_3\, dV}{V} \tag{4}$$

Here, ρ is the mass density. In these problems we consider the mass density to be approximately homogeneous within the egg, so that it cancels out. The centre of mass of the egg is positioned somewhere on the x_3-axis, because this is the symmetry-axis and therefore $x_{1,cm} = x_{2,cm} = 0$ already. As stated before, we want the centre of mass to be at the origin so we impose $x_{3,cm} = 0$ to find the value for $x_{3,\text{shift}}$ in equations (1) and (2). Its value for the example in Figure 1b is $x_{3,\text{shift}} = 3.0$ mm.

2.2 Moments of inertia

Now that the size and mass of the egg are known, we can calculate the principal moments of inertia. These are the moments of inertia about the 1-, 2- and 3-axis, also called the principal axes. Every rotation around an axis perpendicular to the symmetry-axis is symmetrically the same, this means it has the same moment of inertia. Let us choose the following notation as in Analytical Mechanics [Fowless and Cassiday, 2005]: $I_s = I_3$ and $I = I_1 = I_2$. So here I_s is the moment of inertia about the symmetry axis of the egg and I is the moment of inertia about an axis normal to the symmetry axis.

The principal moments of inertia are defined and calculated as follows:

$$I_s = \iiint_V \rho\,(x_1^2 + x_2^2)\, dV \tag{5a}$$

$$I = \iiint_V \rho\,(x_1^2 + x_3^2)\, dV = \iiint_V \rho\,(x_2^2 + x_3^2)\, dV \tag{5b}$$

2.3 Description of the rotation

To describe the rotation of the egg relative to an observer which is standing on the ground watching the egg, we need to specify the orientation of the egg as a function of time in space with respect to a fixed coordinate system. There are different ways to do that, but in this thesis we make use of the Euler angles ϕ, θ and ψ. These angles can describe the orientation of every object in three dimensional space. Let us define the Euler angles in general first and then specify them for the case of an egg. The angles are also illustrated in Figure 2.

We define the fixed coordinate system $Oxyz$, this coordinate system does not rotate with the object and the z-axis is pointing upwards opposite to the gravitational pull. We also have, as already defined for the egg in section 2.1, the $O123$ coordinate system, which consists of the principal axes of the object and thus rotates with the object. When all the Euler angles are zero the objects 3-axis is aligned with the z-axis, the 2-axis with the y-axis and the 1-axis with the x-axis. We can start from this orientation to define step by step the Euler angles (ϕ, θ and ψ) as the following:

1. An anti-clockwise rotation of an angle ϕ around the z-axis, the 1-axis is not anymore aligned with the x-axis now, but with a new axis named x'.

2. An anti-clockwise rotation of an angle θ around the x'-axis, making the x'-axis the intersection between the $x\,y$-plane and the $1\,2$-plane, named 'the line of nodes' [Fowless and Cassiday, 2005].

3. An anti-clockwise rotation of an angle ψ around the 3-axis, so the 1-axis makes an angle ψ with the x'-axis now.

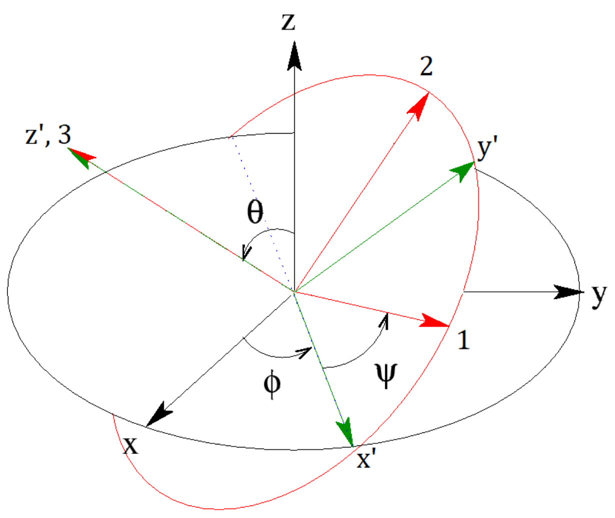


Figure 2: Euler angles ϕ, θ and ψ

There is another coordinate system which can be defined, namely $Ox'y'z'$. Here z' is in the same direction as the 3-axis, the x'-axis is the earlier named line of nodes and the y'-axis is such that we have a orthogonal right-handed coordinate system. The only difference between the $Ox'y'z'$ system and the $O123$ system is that the $Ox'y'z'$ system does not rotates with ψ around the 3-axis.

For the egg the 3-axis (z'-axis) is its symmetry-axis. The 1- and 2-axis can be chosen arbitrarily in the plane perpendicular to the 3-axis (and of course right-handed perpendicular to each other), because, as stated earlier in section 2.2, every rotation around an axis perpendicular to the symmetry-axis is symmetrically the same. So there is no preferred principal 1- and 2-axis in this plane, but when picked they of course rotate with the rotation of the egg.

3 First problem: The freely rotating egg

In this problem we look at an egg that is free to rotate in space, there are no constraints or torques acting on the egg. That is why this problem is known as the freely rotating egg.

3.1 Equation of motion

The Euler angles are defined and the calculation of the moments of inertia is known, now we can start deriving the equation of motion for the case of a freely rotating egg. This can be done in different ways, in this problem we derive it from the fundamental equation of motion in a rotating system [Fowless and Cassiday, 2005]:

$$\mathbf{N} = \left(\frac{d\mathbf{L}}{dt}\right)_{fixed} = \left(\frac{d\mathbf{L}}{dt}\right)_{rot} + \boldsymbol{\omega'} \times \mathbf{L} \tag{6}$$

$\mathbf{N}$ is the torque on the object in the rotating frame, $\mathbf{L}$ is the angular momentum of the object in the rotating frame and $\boldsymbol{\omega'}$ is the angular velocity of the rotating frame.

Let us choose a rotating coordinate system for the egg in which the moments of inertia about the axes do not depend on time. Of course the $O123$ coordinate system will suffice, but we choose a more convenient one, the $Ox'y'z'$ coordinate system. This does not matter for the requirement that the moments of inertia about the axes of the system needs to be time-independent, because, the moment of inertia about any axis perpendicular to the symmetry axis of the egg always has the same value I, as aforementioned.

Now we need to express the torque and angular momentum in the $Ox'y'z'$ system and the angular velocity of the $Ox'y'z'$ system in terms of the Euler angles. The torque is $\mathbf{N} = \mathbf{0}$ here, because we study a freely rotating egg. The angular velocity $\boldsymbol{\omega}$ of the egg is given by $\boldsymbol{\omega} = \dot{\boldsymbol{\phi}} + \dot{\boldsymbol{\theta}} + \dot{\boldsymbol{\psi}}$. While the angular velocity $\boldsymbol{\omega'}$ of the $Ox'y'z'$ system is described only by the rotations $\dot{\boldsymbol{\phi}}$ and $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\omega'} = \dot{\boldsymbol{\phi}} + \dot{\boldsymbol{\theta}}$. The components of $\dot{\boldsymbol{\phi}}$, $\dot{\boldsymbol{\theta}}$ and $\dot{\boldsymbol{\psi}}$ in the $Ox'y'z'$ coordinate system are

$$\begin{aligned} \dot{\phi}_{x'} &= 0 & \dot{\theta}_{x'} &= \dot{\theta} & \dot{\psi}_{x'} &= 0 \\ \dot{\phi}_{y'} &= \dot{\phi}\sin(\theta) & \dot{\theta}_{y'} &= 0 & \dot{\psi}_{y'} &= 0 \\ \dot{\phi}_{z'} &= \dot{\phi}\cos(\theta) & \dot{\theta}_{z'} &= 0 & \dot{\psi}_{z'} &= \dot{\psi} \end{aligned} \tag{7}$$

From this, we get the expression for angular velocity of the $Ox'y'z'$ system as

$$\boldsymbol{\omega'} = \begin{pmatrix} \dot{\theta} \\ \dot{\phi}\sin(\theta) \\ \dot{\phi}\cos(\theta) \end{pmatrix} \tag{8}$$

and the angular velocity of the egg in the $Ox'y'z'$ system as

$$\boldsymbol{\omega} = \begin{pmatrix} \dot{\theta} \\ \dot{\phi}\sin(\theta) \\ \dot{\phi}\cos(\theta) + \dot{\psi} \end{pmatrix} \tag{9}$$

Now we only need the expression for the angular momentum $\boldsymbol{L}$ of the egg in the Ox'y'z' system. This follows from the angular velocity of the egg in this system

$$\boldsymbol{L} = \begin{pmatrix} I_{x'}\,\omega_{x'} \\ I_{y'}\,\omega_{y'} \\ I_{z'}\,\omega_{z'} \end{pmatrix} = \begin{pmatrix} I\dot{\theta} \\ I\dot{\phi}\sin(\theta) \\ I_s S \end{pmatrix} \tag{10}$$

$S = \dot{\phi}\cos(\theta) + \dot{\psi}$ is called the spin of the egg, it is the total angular velocity around the symmetry-axis.

Filling equations (8) and (10) in to the equation of motion (6), gives us the differential equation of the freely rotating egg in terms of our defined Euler angles

$$0 = I\ddot{\theta} + I_s S\dot{\phi}\sin(\theta) - I\dot{\phi}^2\cos(\theta)\sin(\theta) \tag{11a}$$

$$0 = I\frac{d}{dt}(\dot{\phi}\sin(\theta)) - I_s S\dot{\theta} + I\dot{\theta}\dot{\phi}\cos(\theta) \tag{11b}$$

$$0 = I_s\dot{S} \tag{11c}$$

Equation (11c) shows that the angular momentum along the z'-axis (symmetry axis) is constant, so this equation can also be written as

$$L_{z'} = I_s S = \text{constant} \tag{12}$$

multiplying equation (11b) by $\sin(\theta)$ and using equation (12), it follows that

$$0 = \frac{d}{dt}(I\dot{\phi}\sin^2(\theta) + I_s S\cos(\theta)) \tag{13}$$

the term in brackets happens to be exactly the angular momentum around the fixed z-axis from the $Oxyz$ frame, namely $L_z = L_{z'}\cos(\theta) + L_{y'}\sin(\theta)$. So eventually equation (11b) can be written as

$$L_z = I\dot{\phi}\sin^2(\theta) + I_s S\cos(\theta) = \text{constant} \tag{14}$$

Now we are left with only a second-order differential equation for θ, with constants of motion L_z and $L_{z'}$:

$$\begin{aligned}\ddot{\theta} &= \frac{-L_{z'}\dot{\phi}\sin(\theta) + I\dot{\phi}^2\cos(\theta)\sin(\theta)}{I} \\ &= -\,L_{z'}\left(\frac{L_z - L_{z'}\cos(\theta)}{I^2\sin(\theta)}\right) + \left(\frac{(L_z - L_{z'}\cos(\theta))^2}{I^2\sin^3\theta)}\right)\cos(\theta)\end{aligned} \tag{15}$$

here $\dot{\phi}$ and S are filled in, they follow directly from equations (12) and (14)

$$\dot{\phi} = \frac{L_z - L_{z'}\cos(\theta)}{I\sin^2(\theta)} \tag{16}$$

and $\dot{\psi}$ follows from the definition of the spin

$$\dot{\psi} = \frac{L_{z'}}{I_s} - \dot{\phi}\cos(\theta) \tag{17}$$

The three equations (15), (16) and (17) describe the motion of the freely rotating egg. Before numerically solving it, let us develop a bit more understanding of them. In the fixed coordinate system $Oxyz$, the angular momentum $\mathbf{L}$ remains constant in direction and magnitude, because no torque acts on the egg. The direction of the angular momentum vector $\mathbf{L}$ in our case can be determined by of course the components L_z and $L_{z'}$, but also by $L_{x'} = I\dot{\theta}$ which is not a constant. So the direction of the angular momentum vector $\mathbf{L}$ depends on our choice of the constants of motion L_z and L_z, and the initial condition of $\dot{\theta}$. $L_{z'}$ is only constant in magnitude in the fixed coordinate system $Oxyz$, but as it is the component in the z'-direction, it means that the z'-axis is only allowed to rotate with a constant angle around the angular momentum vector $\mathbf{L}$. In our case it means that the symmetry-axis of the egg is only allowed to trace out a cone around the angular momentum vector $\mathbf{L}$. From equation (10), where the angular momentum is expressed in the $Ox'y'z'$ coordinate system, it can be seen that the z'-axis, the angular momentum vector and the angular velocity vector lie in the same plane. Thus also $\boldsymbol{\omega}$ is only allowed to rotate with (another) constant angle around $\mathbf{L}$ with the same frequency as the z'-axis does. The surface traced out by $\boldsymbol{\omega}$ about $\mathbf{L}$ is called the space cone in Analytical Mechanics [Fowless and Cassiday, 2005]. The motion of $\boldsymbol{\omega}$ and the symmetry-axis can be visualized by the 'body cone' rolling along the space cone [Fowless and Cassiday, 2005].

3.2 Description of the numerical code

As stated in section 3, the freely rotating motion of the egg is described by the differential equations (15), (16) and (17). The way these equations are numerically solved is described shortly in this section.

The second order differential equation (15) can be written as two coupled first order differential equations:

$$\frac{d^2\theta}{dt^2} = f(\theta) \ \rightarrow \begin{cases} \dfrac{d\theta}{dt} = \dot{\theta} \\ \dfrac{d\dot{\theta}}{dt} = f(\theta) \end{cases} \tag{18}$$

There are several numerical methods to solve first order differential equations. A popular one, that is also used in this thesis, is the fourth-order Runge-Kutta method [Cheever, 2005]. If we write a coupled set of first-order differential equation as

$$\frac{d\boldsymbol{y}}{dt} = g(\boldsymbol{y}, t) \tag{19}$$

this method is defined as follows:

$$\boldsymbol{y}(t + \Delta t) = \boldsymbol{y}(t) + \frac{1}{6}(k_1 + 2k2 + 2k_3 + k_4) \tag{20}$$

where

$$\begin{aligned}
k_1 &= \Delta t\, g(\boldsymbol{y}, t) \\
k_2 &= \Delta t\, g(\boldsymbol{y} + \frac{1}{2}k_1, t + \frac{1}{2}\Delta t) \\
k_3 &= \Delta t\, g(\boldsymbol{y} + \frac{1}{2}k_2, t + \frac{1}{2}\Delta t) \\
k_4 &= \Delta t\, g(\boldsymbol{y} + k_3, t + \Delta t)
\end{aligned}$$

In our case, we define the equations (15), (16) and (17) as a coupled set of first order differential equation, so

$$\boldsymbol{y}(t) = \begin{pmatrix} \phi(t) \\ \theta(t) \\ \dot{\theta}(t) \\ \psi(t) \end{pmatrix} \quad \text{and} \quad f(\boldsymbol{y}, t) = \begin{pmatrix} \frac{L_z - L_{z'}\cos(\theta)}{I\sin^2(\theta)} \\ \dot{\theta}(t) \\ \frac{-L_{z'}\dot{\phi}\sin(\theta) + I\dot{\phi}^2\cos(\theta)\sin(\theta)}{I} \\ \frac{L_{z'}}{I_s} - \dot{\phi}\cos(\theta) \end{pmatrix} \tag{21}$$

So to solve this differential equation, there is need for the four initial conditions $\boldsymbol{y}(0) = \begin{pmatrix}\phi_0, & \theta_0, & \dot{\theta}_0, & \psi_0\end{pmatrix}^T$ and the constants of motion L_z and $L_{z'}$.

3.3 Simulations

All the useful information to do simulation on a freely rotating egg is explained in previous sections. Let us give the values of the parameters we use and point out what simulations we are going to do in this section.

The values a, b and d as shown in Figure 1a (section 2.1) are used, which were chosen because it represents the ordinary size of a chicken egg. For $a =$ 30 mm, $b = 22$ mm and $d = 5$ mm the calculated volume of the egg, using scipy.integrate in Python, is $V \approx 60.5$ mL. This is the volume of an average European large egg [ix] which has a mass of around $m = 63$ g. From this the mass density is obtained: $\rho = 1041.6$ kg m^{-3}. The calculated value of $x_{3,\text{shift}}$, for shifting the parametrization such that the centre of mass is at the origin, is then $x_{3,\text{shift}} = 3.0$ mm. The moments of inertia I and I_s are determined as in section 2.2. They are, as well calculated using scipy.integrate in Python: $I_s \approx 1.22 * 10^{-5}$ kg m^2 and $I \approx 1.75 * 10^{-5}$ kg m^2.

Two simulations with different rotational initial conditions and constants of motion are performed. One where the angular momentum vector $\mathbf{L}$ points exactly in the fixed z-axis direction, so $L_z = L$, $L_{z'} = L\cos(\theta)$ and $\dot{\theta}_0 = 0$. This because this case is analytically solvable as done in Analytical Mechanics [Fowless and Cassiday, 2005], so we can check whether we get the expected solution. For the other simulation we choose the values of the initial conditions and the constant of motion arbitrarily but physically possible, as if someone throws the egg in the air without a preconceived plan. Because L_z and $L_{z'}$ are in the order of the moments of inertia times a angular velocity, these have to be in the

[ix] Chicken egg sizes: `https://gi-ovo.com/wp-content/uploads/2018/12/Chicken-egg-sizes.pdf`

order of around 10^{-5} kg m^2 s^{-1}, as the calculated moments of inertia.

The chosen values of the initial conditions and the constant of motion of the simulations are:

	Simulation 1	Simulation 2
ϕ_0	0 rad	$\pi/4$ rad
θ_0	$\pi/5$ rad	$\pi/3$ rad
ψ_0	0 rad	0 rad
$\dot{\theta}_0$	0 rad s^{-1}	$\pi/2$ rad s^{-1}
L_z	$6 * 10^{-5}$ kg m^2 s^{-1}	$-14 * 10^{-5}$ kg m^2 s^{-1}
$L_{z'}$	$\cos(\theta_0) L_z \approx 4.0 * 10^{-5}$ kg m^2 s^{-1}	$8 * 10^{-5}$ kg m^2 s^{-1}

Table 1: Rotational initial conditions and constants of motion

In a third simulation we want to give an impression of the full motion, translation and rotation, of the egg relative to an observer standing on earth throwing the egg in the air. There is need for another coordinate system that translates in time with the throw-velocity and the gravity: the coordinate system $Ox_{trans}y_{trans}z_{trans}$. We choose the y_{trans}-axis to be aligned with the throw direction. The $Ox_{trans}y_{trans}z_{trans}$ coordinate system is then defined as follows

$$x_{trans} = x \tag{22a}$$

$$y_{trans} = y + v_0 \cos(\epsilon) t \tag{22b}$$

$$z_{trans} = z + v_0 \sin(\epsilon) t - \frac{1}{2} g t^2 + z_0 \tag{22c}$$

v_0 is the throw-velocity, ϵ is the the angle between the ground and the direction in which the egg is thrown, z_0 is the height at which the thrower let go of the egg, g is the gravitational acceleration and t is the time.
In the third simulation we use the same rotational initial condition and constants of motion as in the second simulation. The chosen values for the translational initial conditions are based on a child throwing an egg underhand into the air, these are given by:

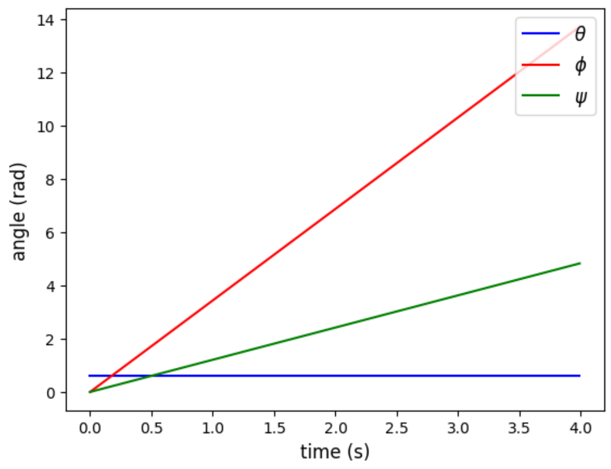
	Simulation 3
v_0	3.2 m s^{-1}
ϵ	$2\pi/5$ rad
z_0	0.8 m

Table 2: Translational initial conditions

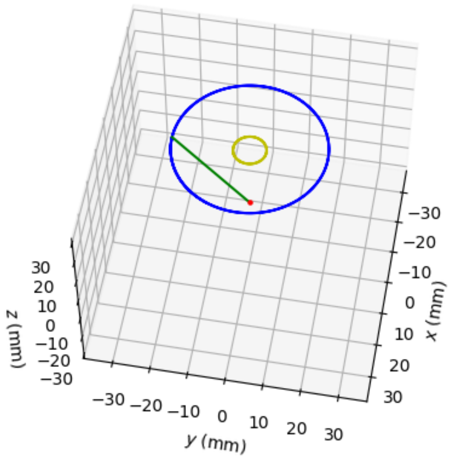
3.4 Results

3.4.1 Simulation 1

The results obtained from the first simulation are expressed in the Figures below. Figure 4 is a snapshot from the animation of the rotating egg, a link to the animation is provided in the caption.



(a) The Euler angles as a function of time.



(b) Position of the tip of the egg during rotation indicated in blue. The red dot is the origin (centre of mass) and the green line indicates the initial state. The yellow line indicates a cross section of the space cone.

Figure 3

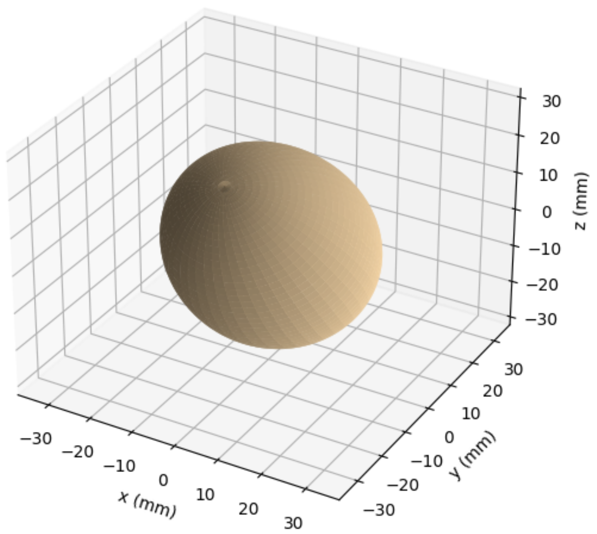
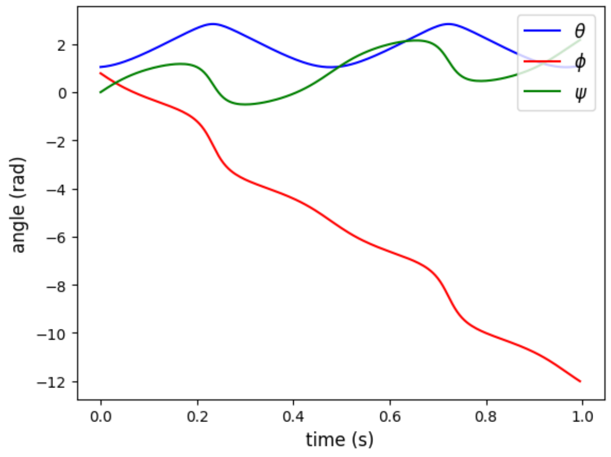


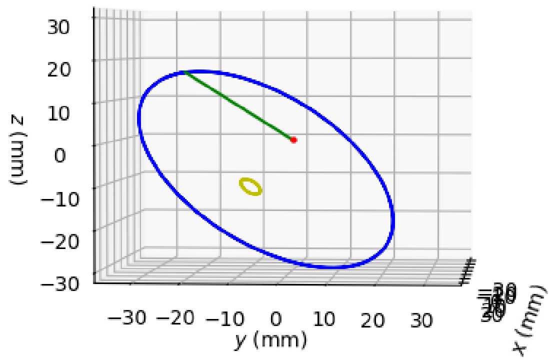
Figure 4: Animation of a freely rotating egg following from simulation 1: `https://youtu.be/zjugc0OFL98`

3.4.2 Simulation 2

The results obtained from the second simulation are expressed in the Figures below. Figure 6 is a snapshot from the animation of the rotating egg, a link to the animation is provided in the caption.



(a) The Euler angles as a function of time



(b) Position of the tip of the egg during rotation indicated in blue. The red dot is the origin (centre of mass) and the green line indicates the initial state. The yellow line indicates a cross section of the space cone.

Figure 5

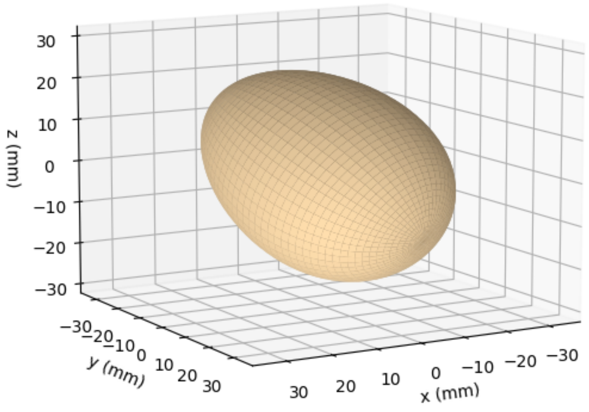


Figure 6: Animation of a freely rotating egg following from simulation 2: `https://youtu.be/nf0MIBTihxc`

3.4.3 Simulation 3

Figure 7 is a snapshot from the animation of the rotating and translating egg, a link to the animation is provided in the caption. The animation is slowed down 5 times to give a better view of the simultaneous rotation and translation of the egg.

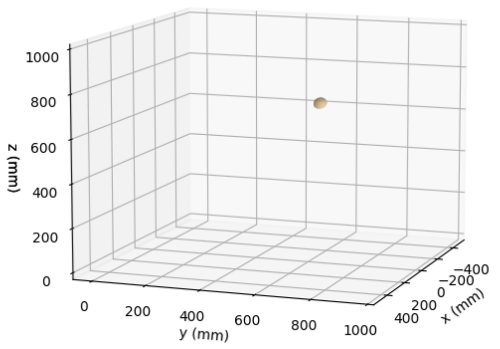


Figure 7: Animation of a freely rotating and translating egg following from simulation 3: `https://youtu.be/84bN6A-oMw8`

3.5 Discussion

From the result of simulation 1, Figure 3a it can be seen that θ stays constant as the symmetry axis (z'-axis) rotates around the angular momentum $\mathbf{L}$ which is, in this simulation, aligned with the z-axis. When θ stays constant, $\dot{\phi}$ and $\dot{\psi}$ are constant as well, following from equations (16) and (17), so ϕ and ψ are growing linearly in time as seen in Figure 3a. The constant rotation around $\mathbf{L}$, given by $\dot{\phi}$, is called precession [Fowless and Cassiday, 2005]. Also the angular velocity $\boldsymbol{\omega}$ is precessing around the angular momentum $\mathbf{L}$, the surface traced out by $\boldsymbol{\omega}$ is known as the space cone as explained in 3.1, the line in yellow in Figure 3b indicates a cross section of the space cone. In the result of simulation 2 the same precessional motion is seen, not around one of the fixed axes, but still around the angular momentum vector $\mathbf{L}$, as visualised in Figure 5b.

In section 3.1 we derived the equation of motion relative to the fixed coordinate system $Oxyz$, where the z-axis points opposite to the gravitational pull. When another fixed coordinate system was chosen, where the z-axis is always aligned with the direction of the angular momentum $\mathbf{L}$ (as in simulation 1), the obtained differential equation would be analytically solvable [Fowless and Cassiday, 2005]. We expect the ability to solve a differential equation analytically cannot depend on the choice of the coordinate system, thus the differential equations (15), (16) and (17) should also be analytically solvable after some substitutions and transformations. One can say that analytically solvable differential equations are preferable to ones that can only be solved numerically, so that makes it a point of discussion. The reason that we settled for the differential equation in our chosen coordinate system is because we otherwise had to define yet another coordinate system, instead of already being in the right coordinate system (from the observer standing on the ground), in which we can solve for different situations. The main reason that we did not solve the differential equation analytically in this thesis has to do with the aim of this project. The aim is to develop an understanding of the motion of an egg, in this case a freely rotating one. This understanding is also achieved when solving the differential equation numerically and we can use the analytical solution given in Analytical Mechanics [Fowless and Cassiday, 2005] to confirm the correctness of this solution. Furthermore, the knowledge obtained in this problem is a very useful stepping stone towards problem two. In that problem there is no possibility for an exact analytical solution, so we can use our numerical method for solving differential equations there.

4 Second problem: Forced rise of the rotating egg

In this problem we look at an egg that is constrained to move on a horizontal plane, the table. It can roll or slide depending on the roughness of the table and it experiences a torque due to gravity. That is why this problem is called 'forced'.

4.1 Lagrange equations

The derivation of the equation of motion in this problem is done using the Lagrangian and the Lagrange equations [Fowless and Cassiday, 2005]. The Lagrangian is defined as

$$L = T_{rot} + T_{trans} - V \tag{23}$$

where T_{rot} is the rotational kinetic energy, T_{trans} the translational kinetic energy and V the potential energy. These energies are independent of the coordinate system and are expressed as:

$$T_{rot} = \frac{1}{2}\boldsymbol{\omega}\cdot\mathbf{I}\cdot\boldsymbol{\omega} \tag{24a}$$

$$T_{trans} = \frac{1}{2}mv_{cm}^2 \tag{24b}$$

$$V = mgh \tag{24c}$$

The angular velocity $\boldsymbol{\omega}$ in the $Ox'y'z'$ coordinate system is expressed in equation (9) from section 3.1. The corresponding moment of inertia tensor $\mathbf{I}$ in this coordinate system is defined as

$$\mathbf{I} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I_s \end{pmatrix} \tag{25}$$

so T_{rot} can already be expressed.

The expression of the centre of mass velocity v_{cm} still needs to be found in a certain coordinate system, this is explained in section 4.4. There is also need for an expression of the height of the centre of mass above the table h, this follows first from section 4.3. When all these expressions are found the full Lagrangian is expressed in section 4.5, from which the full equation of motion will be derived.

4.2 Ellipsoid

In this problem the egg is slightly simplified to an ellipsoid, so $d = 0$ in equations (1) and (2) from section 2.1. The parametrization is then given by

$$x_1 = b\sin(\alpha)\cos(\beta) \tag{26a}$$

$$x_2 = b\sin(\alpha)\sin(\beta) \tag{26b}$$

$$x_3 = a\cos(\alpha) \tag{26c}$$

$$\text{where} \quad \alpha = [0,\pi\rangle\ ,\ \beta = [0,2\pi\rangle$$

The main reason of this simplification is that the rising effect also occurs when spinning an ellipsoid. The rise has nothing to do with the shift of symmetry of the ellipsoid to the egg. This is tested by spinning objects with a symmetrical 1 2-plane, like (most of the) small chocolate Easter eggs one can buy during Easter in the supermarket. By doing so, the effect for the ellipsoid can be derived first, before potentially generalizing it to more complicated objects.

The ellipsoid makes thing a lot easier, because now there exists an analytical expression of the, earlier named, height of the centre of mass above the table h, so there is no need for estimating or fitting as there would be if we had the egg shape.

4.3 Contact point

The height of the centre of mass above the table h is the vertical distance from the contact point of the egg with the table to the center of mass of the egg. To find this vertical distance, we need an expression of the contact point. Let us call the contact point P. There is need for one other coordinate system to find an expression for the contact point P. Let us call this coordinate system $OXYZ$.

The YZ plane of the $OXYZ$ coordinate system lies in the same plane as the $y'z'$ plane from the $Ox'y'z'$ coordinate system. The difference is that the $OXYZ$ coordinate system rotates with the Euler angle ϕ only and not with θ as $Ox'y'z'$. In the Figure 8 below the $OXYZ$ coordinate system is illustrated with respect to the $Ox'y'z'$ coordinate system by only showing the overlapping coordinate plane. The contact point P is indicated as well.

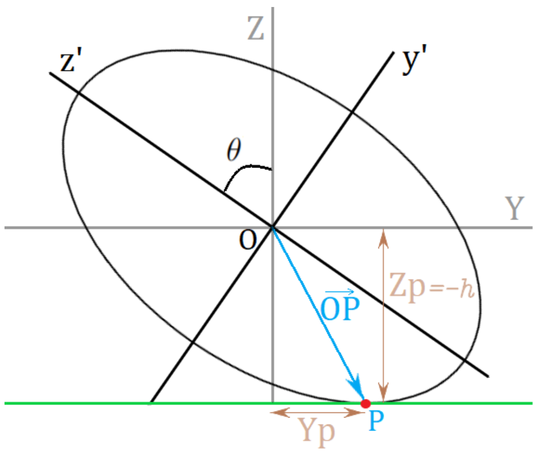


Figure 8: YZ- and $y'z'$-plane of the ellipsoid, P indicates the contact point with the table

The parametrization of the ellipse in the $y'\,z'$-plane is given by:

$$y' = b\sin(\alpha) \tag{27a}$$
$$z' = a\cos(\alpha) \tag{27b}$$
$$\text{where} \quad \alpha = [0, 2\pi\rangle$$

in the $Y\,Z$-plane the parametrization is then given by a rotation over θ:

$$Y = y'(\alpha)\cos(\theta) - z'(\alpha)\sin(\theta) = b\sin(\alpha)\cos(\theta) - a\cos(\alpha)\sin(\theta) \tag{28a}$$
$$Z = y'(\alpha)\sin(\theta) + z'(\alpha)\cos(\theta) = b\sin(\alpha)\sin(\theta) + a\cos(\alpha)\cos(\theta) \tag{28b}$$

In the $OXYZ$ frame the contact point P is given by:

$$\overrightarrow{OP}_{OXYZ} = \begin{pmatrix} 0 \\ Y_P \\ Z_P \end{pmatrix} \tag{29}$$

where Z_P can be expressed as the earlier named, height of the centre of mass, $Z_P = -h$.

To find an expression of the contact point in terms of θ, one needs to minimize the expression of Z as in equation (28) with respect to parametrization angle α. This minimum can be found by setting

$$\left.\frac{dZ}{d\alpha}\right|_{\alpha=\alpha_P} = 0\,, \qquad \left.\frac{d^2Z}{d\alpha^2}\right|_{\alpha=\alpha_P} > 0 \tag{30}$$

from this we get

$$b\cos(\alpha_P)\sin(\theta) - a\sin(\alpha_P)\cos(\theta) = 0 \tag{31}$$

and after some algebra, where we take into account that $\left.\frac{d^2Z}{d\alpha^2}\right|_{\alpha=\alpha_P} > 0$, we are left with

$$\sin(\alpha_P) = \frac{-b\sin(\theta)}{(a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}}} \tag{32a}$$

$$\cos(\alpha_P) = \frac{-a\cos(\theta)}{(a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}}} \tag{32b}$$

fill this in to equation (28b), to find

$$h = -Z_P = -b\frac{-b\sin(\theta)}{(a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}}}\sin(\theta) - a\frac{-a\cos(\theta)}{(a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}}}\cos(\theta) \tag{33}$$

from which we get the elegant equation

$$h(\theta) = (a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}} \tag{34}$$

The expression for Y_P can also be obtained now, by filling equations (32a) and (32b) in equation (28a). After some algebra this becomes

$$Y_P(\theta) = \frac{1}{2}\frac{(a^2 - b^2)\sin(2\theta)}{(a^2\cos^2(\theta) + b^2\sin^2(\theta))^{\frac{1}{2}}} \tag{35}$$

from this we can seen that

$$Y_P(\theta) = -\frac{dh}{d\theta} \tag{36}$$

This could have be obtained earlier by the fact that $\frac{dh}{d\theta} = -\left.\frac{dZ}{d\theta}\right|_{\alpha=\alpha_P}$ and using equation (28) and (30).

Eventually the contact point in the $OXYZ$ coordinate system from equation (29) is written as

$$\overrightarrow{OP}_{OXYZ} = \begin{pmatrix} 0 \\ -\frac{dh}{d\theta} \\ -h \end{pmatrix} \tag{37}$$

with h given in equation (34).

4.4 Partial slip

The rise of the ellipsoid results from the frictional force between the ellipsoid and the table. When the ellipsoid is spun, this force causes a torque which apparently lift the ellipsoid. To model this frictional force let us introduce partial slip. This means the ellipsoid is not perfectly rolling and not only slipping. By perfect rolling is meant that the translational velocity of the object is exactly opposite to the rotational velocity at the contact point of the object with the table, so the net velocity of the objects contact point is zero relative to the table. We define 'perfect slipping'. By this is meant that there is no relation between the rotational velocity of the objects contact point and the translational velocity of the object, the object is only slipping over the perfectly smooth surface. In our case, the ellipsoid is only given rotational initial conditions, this means that when perfectly slipping the object translational velocity components parallel to the table would remain zero.

For partial slipping, the total velocity of the ellipsoids contact point $\mathbf{v_P}$ is unequal to zero and unequal to only rotational velocity, which means unequal to perfect rolling and unequal to perfect slipping respectively. The total velocity is somewhere in between. The translational velocity is the centre of mass velocity $\mathbf{v_{cm}}$. The rotational velocity at the contact point follows from the angular velocity $\boldsymbol{\omega}$ and the location of the contact point $\overrightarrow{OP}$. The frictional force only has components in the X and Y direction so in this directions the partial slipping occurs. We define the partial slipping of the ellipsoid in the $OXYZ$ coordinate system as

$$\mathbf{v_P} = \begin{pmatrix} v_{cm,X} \\ v_{cm,Y} \\ v_{cm,Z} \end{pmatrix} + \boldsymbol{\omega}\times\overrightarrow{OP} = \begin{pmatrix} \mu\,(\boldsymbol{\omega}\times\overrightarrow{OP})_X \\ \mu\,(\boldsymbol{\omega}\times\overrightarrow{OP})_Y \\ 0 \end{pmatrix} \tag{38}$$

so

$$\begin{pmatrix} v_{cm,X} \\ v_{cm,Y} \\ v_{cm,Z} \end{pmatrix} = \begin{pmatrix} (\mu-1)\,(\boldsymbol{\omega}\times\overrightarrow{OP})_X \\ (\mu-1)\,(\boldsymbol{\omega}\times\overrightarrow{OP})_Y \\ -(\boldsymbol{\omega}\times\overrightarrow{OP})_Z \end{pmatrix} \tag{39}$$

The parameter μ is defined as the slipping parameter. When $\mu = 0$ perfect rolling occurs and when $\mu = 1$ perfect slipping occurs. For our definition of partial slipping, the value of μ is somewhere between 0 and 1.

Now there is need for an expression of $\boldsymbol{\omega}$ in the $OXYZ$ coordinate system. This expression is obtained from equation (9)

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_{x'} \\ \omega_{y'}\cos(\theta) - \omega_{z'}\sin(\theta) \\ \omega_{y'}\sin(\theta) + \omega_{z'}\cos(\theta) \end{pmatrix} = \begin{pmatrix} \dot{\theta} \\ (\dot{\phi}\cos(\theta) - S)\sin(\theta) \\ \dot{\phi}\sin^2(\theta) + S\cos(\theta) \end{pmatrix} \tag{40}$$

filling this and equation (37) in equation (39) gives us

$$\begin{pmatrix} v_{cm,X} \\ v_{cm,Y} \\ v_{cm,Z} \end{pmatrix} = \begin{pmatrix} (\mu-1)\,\left((S - \dot{\phi}\cos(\theta))\sin(\theta)h + (\dot{\phi}\sin^2(\theta) + S\cos(\theta))\frac{dh}{d\theta}\right) \\ (\mu-1)\,\dot{\theta}h \\ \dot{\theta}\frac{dh}{d\theta} \end{pmatrix} \tag{41}$$

4.5 Equation of motion

From the above sections, all terms have been derived to describe the complete Lagrangian

$$\begin{aligned} L = &\frac{1}{2}(I\dot{\theta}^2 + I(\dot{\phi}\sin(\theta))^2 + I_s S^2) \\ &+ \frac{1}{2}m\Bigg(\dot{\theta}^2\Big((\mu-1)^2 h^2 + \Big(\frac{dh}{d\theta}\Big)^2\Big) \\ &\qquad + (\mu-1)^2\Big((S-\dot{\phi}\cos(\theta))\sin(\theta)h + (\dot{\phi}\sin^2(\theta)+S\cos(\theta))\frac{dh}{d\theta}\Big)^2\Bigg) \\ &- mgh \end{aligned} \tag{42}$$

from this the equations of motion are derived using the Lagrange equations. With the Euler angles as the generalized coordinates, the Lagrange equations are defined as follows

$$\frac{dL}{d\theta} = \frac{d}{dt}\left(\frac{dL}{d\dot{\theta}}\right) \tag{43a}$$

$$\frac{dL}{d\phi} = \frac{d}{dt}\left(\frac{dL}{d\dot{\phi}}\right) \tag{43b}$$

$$\frac{dL}{d\psi} = \frac{d}{dt}\left(\frac{dL}{d\dot{\psi}}\right) \tag{43c}$$

Equation (43a) yields the differential equation for θ

$$\begin{aligned} \ddot{\theta}&\left(I + mh^2(\mu-1)^2 + m\left(\frac{dh}{d\theta}\right)^2\right) = \\ &\qquad - \dot{\theta}^2 m\left(h\frac{dh}{d\theta}(\mu-1)^2 + \frac{dh}{d\theta}\frac{d^2h}{d\theta}\right) + I\dot{\phi}^2\sin(\theta)\cos(\theta) \\ &\qquad\qquad - I_S S\dot{\phi}\sin(\theta) + mv_{cm,X}\frac{dv_{cm,X}}{d\theta} - mg\frac{dh}{d\theta} \end{aligned} \tag{44}$$

where $v_{cm,X}$ given in equation (41) and his derivative to θ given by

$$\frac{dv_{cm,X}}{d\theta} = (\mu-1)\left((S-\dot{\phi}\cos(\theta))\cos(\theta)h + (\dot{\phi}\sin^2(\theta)+S\cos(\theta))\frac{d^2h}{d\theta^2}\right) \tag{45}$$

From equations (43b) and (43c) the constants of motion are derived as

$$\frac{dL}{d\dot{\phi}} = I\dot{\phi}\sin^2(\theta) + I_s S\cos(\theta) + mv_{cm,X}(\mu-1)\frac{dh}{d\theta} = \text{constant} = L_z \tag{46}$$

$$\frac{dL}{d\dot{\psi}} = I_s S + mv_{cm,X}(\mu-1)\Big(\cos(\theta)\frac{dh}{d\theta} + \sin(\theta)h\Big) = \text{constant} = L_{z'} \tag{47}$$

As $\dot{\phi}$ and $\dot{\psi}$ indicates a rotation around the z-axis and the z'-axis respectively, these constants of motion are the components of the angular momentum $\mathbf{L}$ along these axes.

Equations (46) and (47) can be written as

$$f_{11}(\theta)\dot{\phi} + f_{12}(\theta)S = L_z \tag{48a}$$

$$f_{21}(\theta)\dot{\phi} + f_{22}(\theta)S = L_{z'} \tag{48b}$$

where the functions f_{ij} are given by:

$$f_{11}(\theta) = I\sin^2(\theta) + m(\mu-1)^2\frac{dh}{d\theta}\Big(\sin^2(\theta)\frac{dh}{d\theta} - \cos(\theta)\sin(\theta)h\Big) \tag{49a}$$

$$f_{12}(\theta) = I_s\cos(\theta) + m(\mu-1)^2\frac{dh}{d\theta}\Big(\sin(\theta)h + \cos(\theta)\frac{dh}{d\theta}\Big) \tag{49b}$$

$$f_{21}(\theta) = m(\mu-1)^2\Big(\sin(\theta)h + \cos(\theta)\frac{dh}{d\theta}\Big)\Big(\sin^2(\theta)\frac{dh}{d\theta} - \cos(\theta)\sin(\theta)h\Big) \tag{49c}$$

$$f_{22}(\theta) = I_s + m(\mu-1)^2\Big(\sin(\theta)h + \cos(\theta)\frac{dh}{d\theta}\Big)^2 \tag{49d}$$

Now we can write $\dot{\phi}$ and S as a function of θ only

$$\dot{\phi} = \frac{L_z f_{22}(\theta) - L_{z'}f_{12}(\theta)}{f_{22}(\theta)f_{11}(\theta) - f_{21}(\theta)f_{12}(\theta)} \tag{50}$$

$$S = \frac{L_{z'}f_{11}(\theta) - L_z f_{21}(\theta)}{f_{22}(\theta)f_{11}(\theta) - f_{21}(\theta)f_{12}(\theta)} \tag{51}$$

4.6 Description of the numerical code

Similar as in problem 1, we have a second order differential equation for θ (now given in equation (44)) from which we obtain $\theta(t)$. From this we get $\dot{\phi}(t)$ and $\dot{\psi}(t)$ (see equations (50) and (51)), from which then $\phi(t)$ and $\psi(t)$ can be obtained. Additionally $\mathbf{v}_{cm}(t)$ can be derived from this (see equation (41)), from which the position of the center of mass $\mathbf{x}_{cm}(t)$ can be obtained. In the numerical code, all this is done in one coupled differential equation, similar as described in problem 1 section 3.2. Here we also make use of the fourth-order Runge-Kutta method for solving this coupled differential equation. To solve the equation, there is need for the initial conditions $\boldsymbol{y}(0) = \Big(\theta_0,\ \dot{\theta}_0,\ \phi_0,\ \psi_0,\ x_0,\ y_0,\ z_0\Big)^T$ and the constants of motion L_z and $L_{z'}$.

4.7 Simulations

Let us now solve the derived equations numerically and simulate the motion of the ellipsoid. First we need to specify the values a an b for the size of the ellipsoid. We choose this values the same as in problem 1, so the size of the ellipsoid is around the size of a chicken egg: $a = 30$ mm an $b = 22$ mm ($d = 0$ now). From this the volume is calculated as $V = \frac{4}{3}\pi b^2 a \approx 6.08 * 10^{-5}$ m^3. Let us take the same mass as used in problem 1, so $m = 63$ g. From this the mass density is obtained: $\rho = 1035.8$ kg. m^{-3}. The moments of inertia I and I_s are determined as in section 2.2. They are, calculated using scipy.integrate in Python: $I_s \approx 1.22 * 10^{-5}$ kg m^2 and $I \approx 1.74 * 10^{-5}$ kg m^2.

After doing so, the initial values and constants of motion can be chosen for our simulations. First, three test-simulations are done to check whether the ellipsoid moves as expected with the given initial conditions and constants. Secondly, three simulations are done for which we predict the ellipsoid to perform the rising effect. In all simulation the value of the following initial conditions are: $\psi_0 = 0$, $x_0 = 0$, $y_0 = 0$ and $z_0 = 0$. The other values are given in the table below. Because L_z and $L_{z'}$ are in the order of the moments of inertia times a angular velocity, these have to be in the order of around 10^{-5} kg m^2 s^{-1}, as the calculated moments of inertia.

	Simulation 1	Simulation 2	Simulation 3
ϕ_0	0 rad	0 rad	0 rad
θ_0	$\pi/10$ rad	$\pi/3$ rad	$\pi/3$ rad
$\dot{\theta}_0$	0 rad s^{-1}	0 rad s^{-1}	0 rad s^{-1}
L_z	0 kg m^2 s^{-1}	$1 * 10^{-5}$ kg m^2 s^{-1}	$1 * 10^{-5}$ kg m^2 s^{-1}
$L_{z'}$	0 kg m^2 s^{-1}	$2 * 10^{-4}$ kg m^2 s^{-1}	$2 * 10^{-4}$ kg m^2 s^{-1}
μ	0	0	0.9

Table 3: Rotational initial conditions and constants of motion of three test simulations

	Simulation 4	Simulation 5	Simulation 6
ϕ_0	0 rad	$\pi/4$ rad	$\pi/4$ rad
θ_0	$\pi/2 + \pi/40$ rad	$\pi/2 + \pi/40$ rad	$\pi/2 + \pi/40$ rad
$\dot{\theta}_0$	0 rad s^{-1}	0 rad s^{-1}	0 rad s^{-1}
L_z	$5.5 * 10^{-4}$ kg m^2 s^{-1}	$1 * 10^{-3}$ kg m^2 s^{-1}	$1 * 10^{-3}$ kg m^2 s^{-1}
$L_{z'}$	0 kg m^2 s^{-1}	0 kg m^2 s^{-1}	0 kg m^2 s^{-1}
μ	0	0	0.5

Table 4: Rotational initial conditions and constants of motion of three simulations for which the rising effect is predicted

Simulation 1 represents an ellipsoid that is standing close to its unstable equilibrium-point, $\theta = 0$, with no components of the angular momentum in the z- and z'-direction, $L_z = L_{z'} = 0$. The value of the slipping parameter is chosen, $\mu = 0$, so perfect rolling should occur. We expect θ to oscillate around the ellipsoids stable equilibrium-point, $\theta = \pi/2$, when this happens of course also the height of the center of mass, z_{cm}, oscillates. Because the ellipsoid is perfectly rolling, the ellipsoid should also translate in the y-direction, so we expect an oscillation of y_{cm} as well.

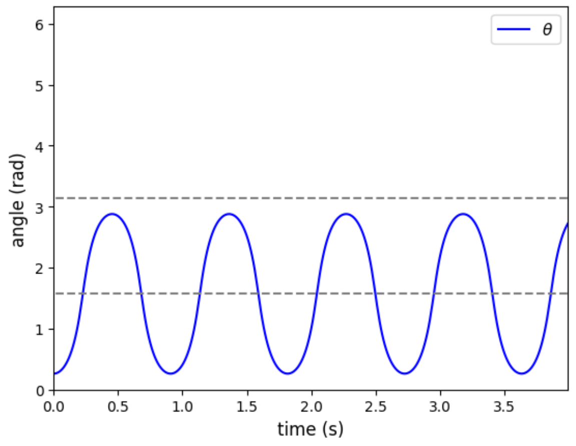
Simulation 2 represents an more advanced situation, but still perfect rolling $\mu = 0$. In this simulation we still expect θ to oscillate around the ellipsoids stable equilibrium-point $\theta = \pi/2$, but with smaller oscillations because the value is chosen closer to the ellipsoids stable equilibrium-point. Because $L_{z'}$ is larger than L_z, we expect the ellipsoid to roll around its symmetry-axis faster and therefore start translating in the $-x$-direction. Since the angular momentum also has a component on the z-axis, it is expected that the rolling and oscillating ellipsoid makes an anti-clockwise turn around the z-axis.

Simulation 3 is the same as simulation 2 except that the value of the slipping parameter is $\mu = 0.9$. This simulation tests the effects of partially slipping on the movement of the ellipsoid. If $\mu = 1$ (perfect slipping), we would expect the ellipsoid to not translate. In this simulation we expect the ellipsoid to slip and translate, but less translation then in simulation 2. We also expect θ to oscillate faster around the ellipsoids stable equilibrium-point $\theta = \pi/2$, then in simulation 2, because the ellipsoid is slipping over the table more, so less 'energy' is lost in pushing the ellipsoid to also translate.

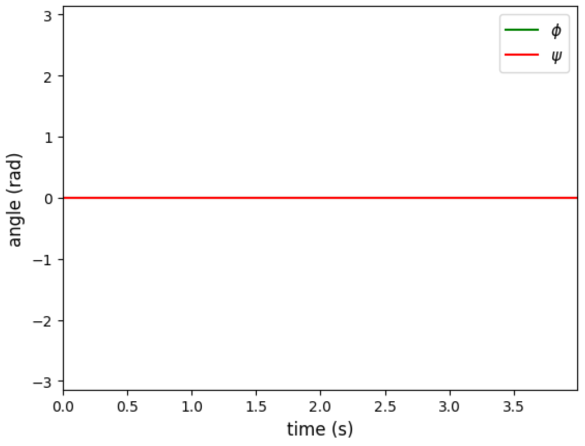
In the other three simulations 4,5 and 6 we try to adapt the initial conditions and constants of motion such that the rising effect should occur. So in all these simulations the ellipsoid is lying with only a small offset from $\theta = \pi/2$. The angular momentum component in the z-direction has a non-zero value and the component in the z'-direction has a value of zero. In simulation 4 and 5 the slipping parameter $\mu = 0$. The difference between these simulations is that the value of L_z in simulation 5 is a lot greater, we expect the ellipsoid to rise faster in this case. Simulation 6 is almost the same as simulation 5, except that the value of the slipping parameter $\mu = 0.5$. When the rising occurs, $\dot{\psi}$ should grow as the ellipsoid starts rotating around its symmetry-axis.

4.8 Results

4.8.1 Simulation 1

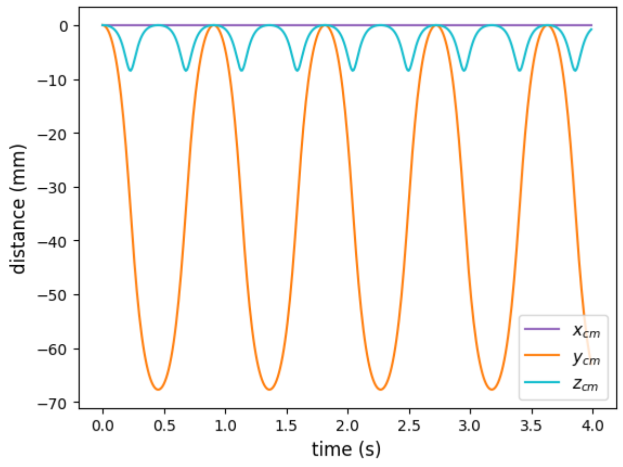


(a) Euler angle θ as a function of time

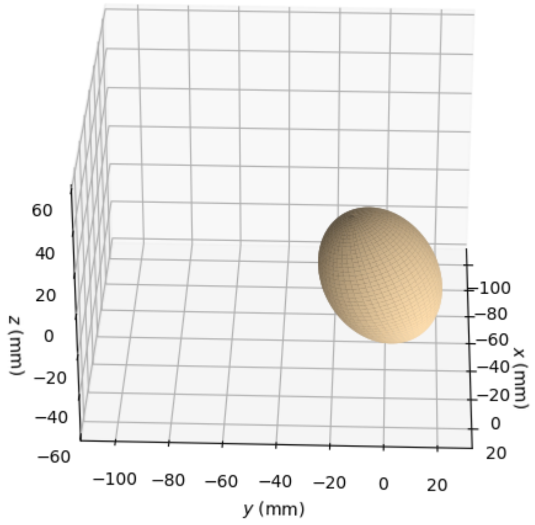


(b) Euler angles ϕ and ψ as a function of time

Figure 9



(a) Translational displacement components in the $Oxyz$ coordinate system

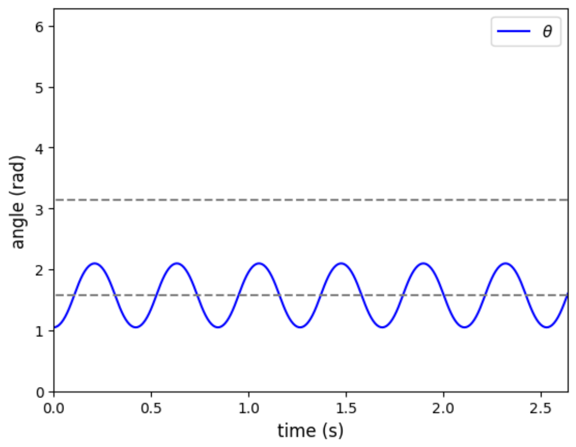


(b) Animation of a perfect rolling ellipsoid following from simulation 1: `https://youtu.be/u0qfvpkoUc8`

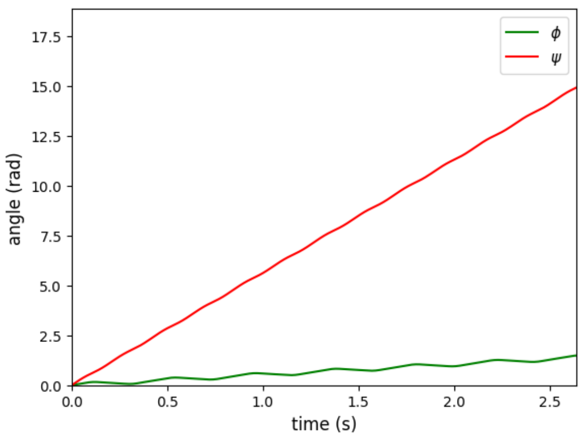
Figure 10

This simulation runs for a time of 4 seconds. As you can see, the ellipsoid acts as we expected.

4.8.2 Simulation 2

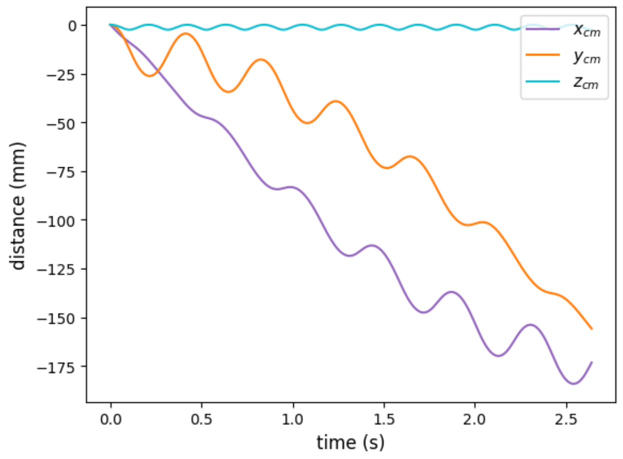


(a) Euler angle θ as a function of time

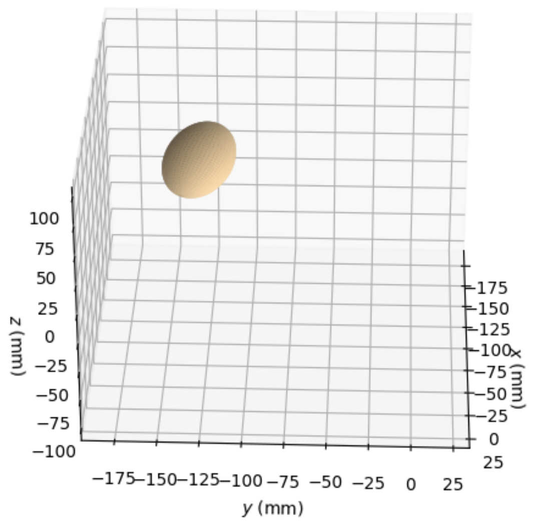


(b) Euler angles ϕ and ψ as a function of time

Figure 11



(a) Translational displacement components in the $Oxyz$ coordinate system

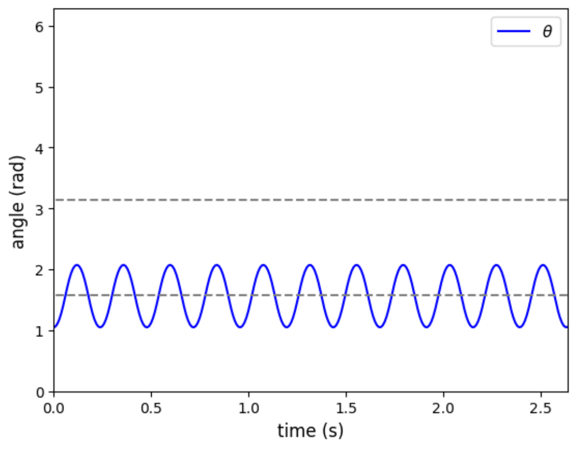


(b) Animation of a perfect rolling ellipsoid following from simulation 2: `https://youtu.be/IrepnPLFlvU`

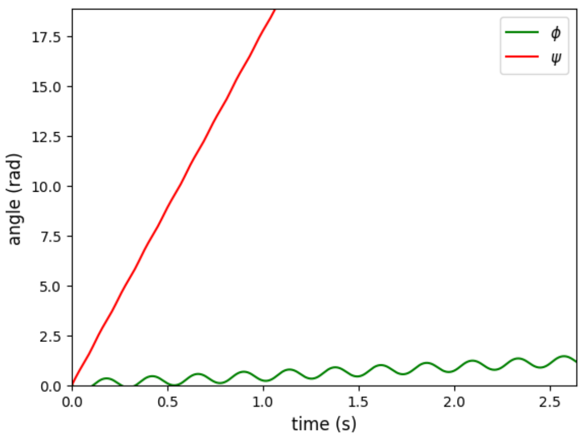
Figure 12

This simulation runs for a time of 2.7 seconds. As you can see, the ellipsoid acts as we expected.

4.8.3 Simulation 3

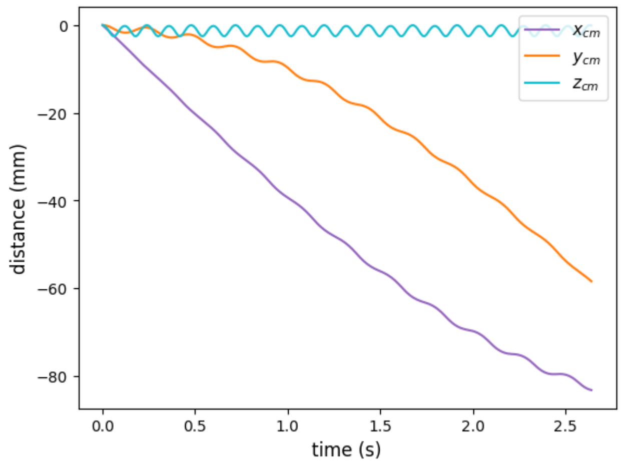


(a) Euler angle θ as a function of time

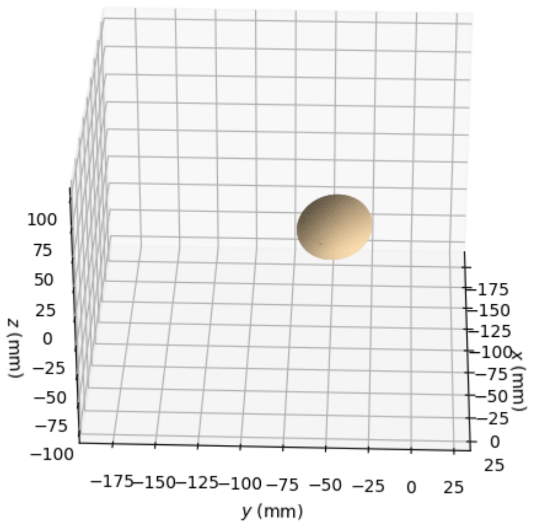


(b) Euler angles ϕ and ψ as a function of time

Figure 13



(a) Translational displacement components in the $Oxyz$ coordinate system

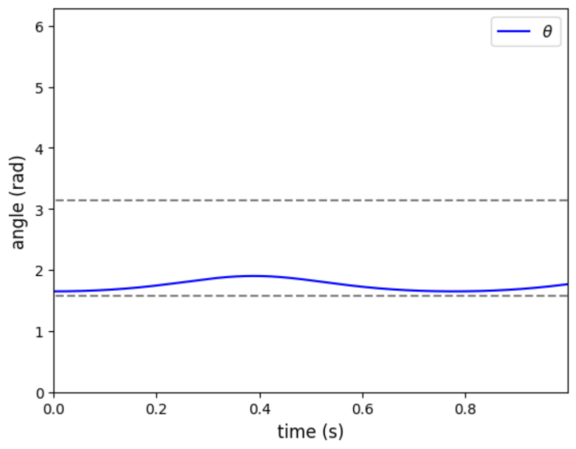


(b) Animation of a partially slipping ellipsoid following from simulation 3:
`https://youtu.be/kIBBUll1wko`

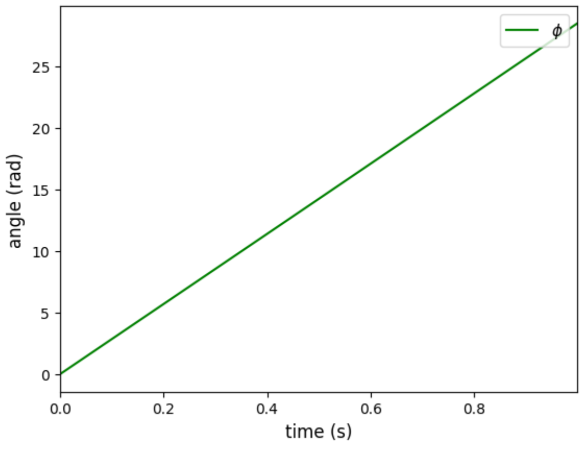
Figure 14

This simulation runs for a time of 2.7 seconds. As you can see, the ellipsoid acts as we expected.

4.8.4 Simulation 4

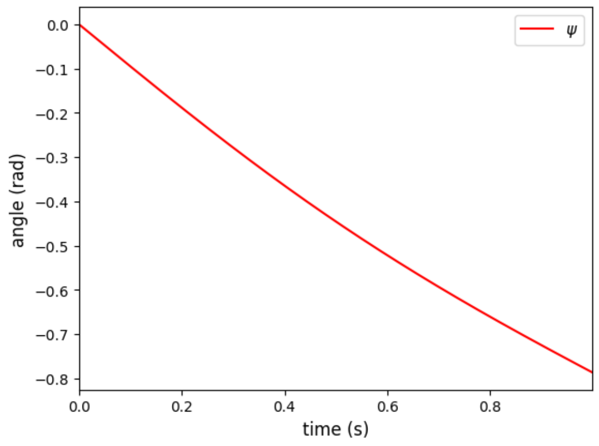


(a) Euler angle θ as a function of time

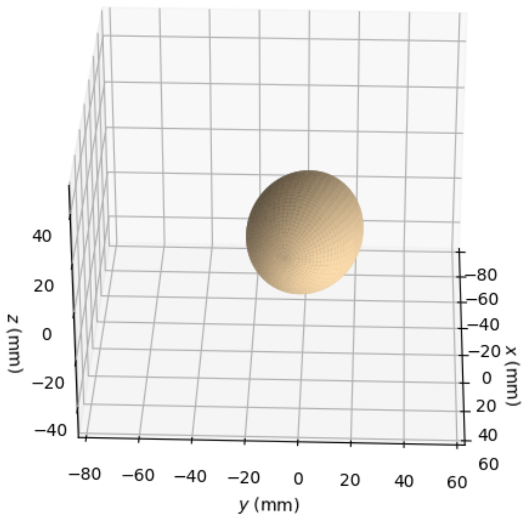


(b) Euler angle ϕ as a function of time

Figure 15



(a) Euler angle ψ as a function of time

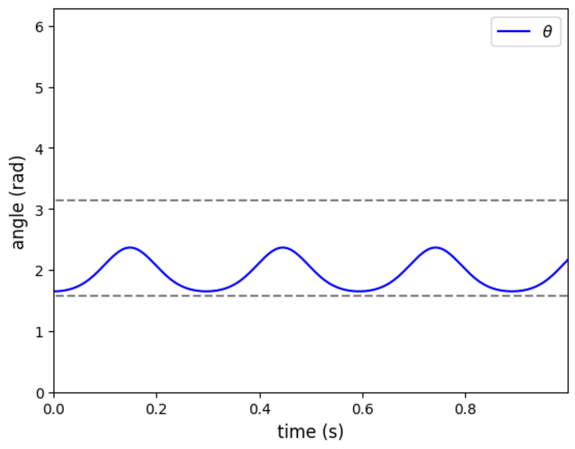


(b) Animation of a rotating ellipsoid following from simulation 4. The time is slowed down 3 times in this animation:
`https://youtu.be/pEIgdPF4sAo`

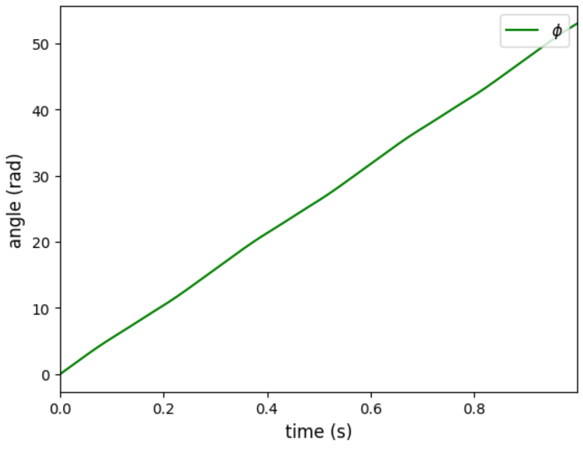
Figure 16

This simulation runs for a time of 1 second. As you can see, the ellipsoid does not oscillate around $\theta = \pi/2$ and some of the rising occurs. But the ellipsoid keeps falling back. The value of the Euler angle ψ behaves not as we expect, it is grows very slow in the negative direction, which means a clockwise rotation around the symmetry-axis, while we expect in to grow in the same direction as ϕ, an anti-clockwise rotation around the z-axis. Compared to the fast rotation of ϕ, the rotation of ψ from this result is negligible.

4.8.5 Simulation 5

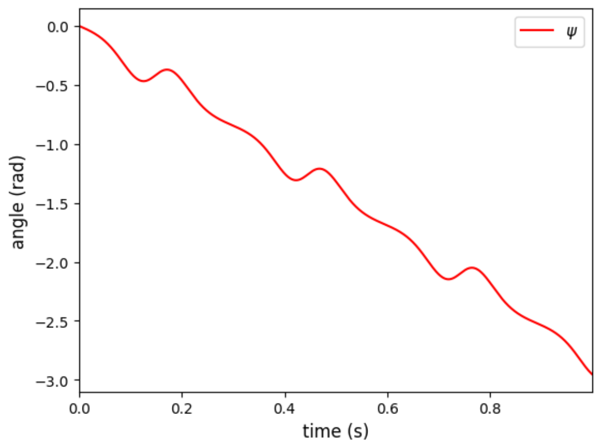


(a) Euler angle θ as a function of time

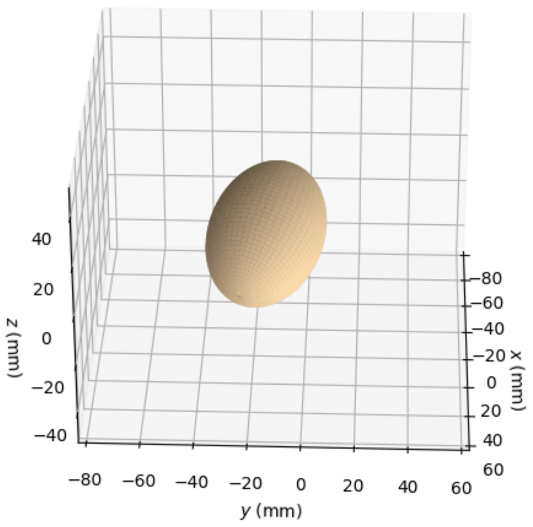


(b) Euler angle ϕ as a function of time

Figure 17



(a) Euler angle ψ as a function of time

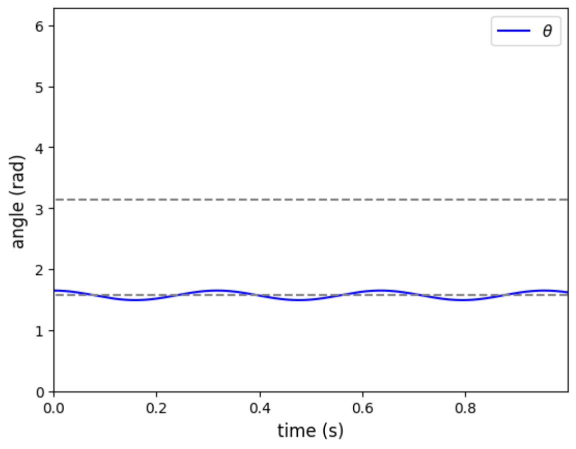


(b) Animation of a rotating ellipsoid following from simulation 5. The time is slowed down 3 times in this animation: `https://youtu.be/xYQUHNKOEA0`

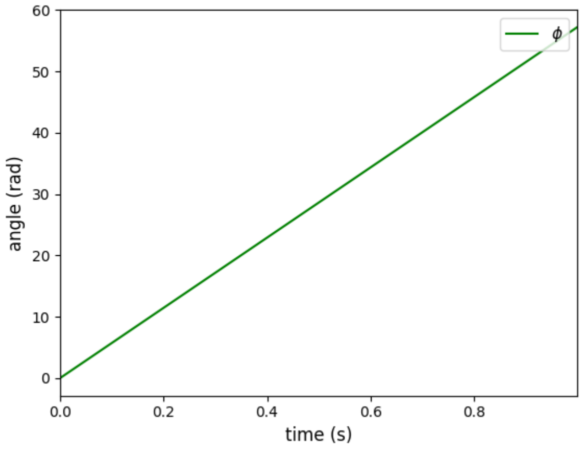
Figure 18

This simulation runs for a time of 1 second. As you can see, the ellipsoid does not oscillate around $\theta = \pi/2$ here. It rises higher then in simulation 4. But the ellipsoid keeps falling back in this simulation as well. Again the Euler angle ψ does not behave as we expect. The rotation of ψ from this result is still negligible compared to the fast rotation of ϕ.

4.8.6 Simulation 6

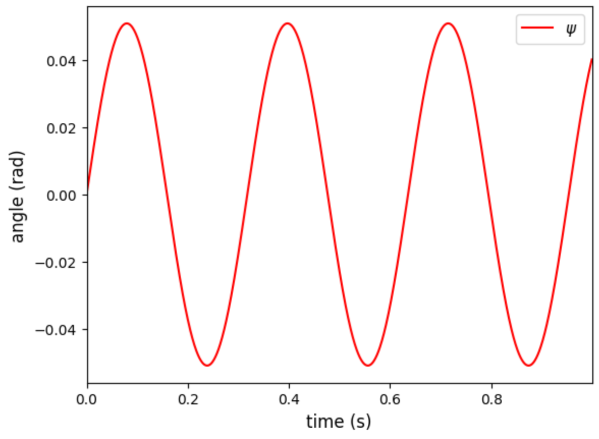


(a) Euler angle θ as a function of time

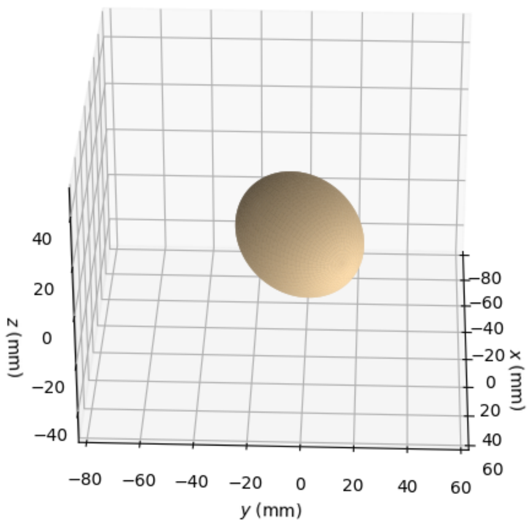


(b) Euler angle ϕ as a function of time

Figure 19



(a) Euler angle ψ as a function of time



(b) Animation of a rotating ellipsoid following from simulation 6. The time is slowed down 3 times in this animation: `https://youtu.be/752jzOJwmX0`

Figure 20

This simulation runs for a time of 1 second. As you can see, the ellipsoid does oscillate around $\theta = \pi/2$ here. There is no rising effect. The value of the Euler angle ϕ is growing faster then in simulation 5.

4.9 Discussion

The results of simulations 1, 2 and 3 are in accordance with the expectations listed in section 4.7. The perfect rolling happens in simulation 1 and 2 as expected and when the slipping parameter μ is close to 1, as in simulation 3, the ellipsoid starts slipping and less translation happens. But with the value of μ between 0 and 1, the ellipsoid is not slowed down during its rolling and slipping, while in reality this slowing down would happen due to the friction between the ellipsoid and the table. That this does not happen in our model has to do with the defined Lagrangian. In the Lagrangian we modeled the frictional force as a constant constraint on the velocity of the contact point of the ellipsoid. This constant constraint does not model the dynamical friction. Dynamical friction is not absorbable in the Lagrangian in this way.

The results from simulation 4 and 5 show some of the rising effect, but the ellipsoid keeps falling back. Let us discuss why the full rising effect does not occur in our simulations. It can directly be seen from our constants of motion L_z and $L_{z'}$ that the full rising is not possible in our model. When the ellipsoid is standing, so $\theta = 0$ or $\theta = \pi$, the z-axis and the z'-axis are aligned. As the constants of motion are components of the angular momentum along these axes, their absolute value have to be the same in this situation. As they are 'constants' of motion, they do not change, so for example in simulation 5, $L_{z'}$ stays zero and as L_z has a non-zero value here, the z-axis and the z'-axis are not allowed to align. This is a direct explanation of why the full rising effect cannot occur following our model, there are too many conserved quantities.

5 Conclusions

In this thesis we looked at two different situation of a spinning egg. For the two situations the equations of motion are derived and different simulations are performed. For both problems the results are obtained and discussed. An overall conclusion is drawn in this section.

First a freely rotating egg is studied. It has been identified that in this case the rotation and translation of an egg can be perfectly described by numerically solving the equation governing its motion. As mentioned at the bottom of section 3.1, from the constants of motion L_z and $L_{z'}$, it could be obtained that the symmetry-axis is only allowed to rotate with a constant angle around the angular momentum vector $\mathbf{L}$. Thus these constants of motion reduce the degrees of freedom of the rotation and already give us an understanding of the rotational movement.

In the second problem we looked at a spinning egg constrained to move on a table and experiencing frictional force depending on the roughness of the table. The friction is modeled as a constant constraint on the velocity of the contact point of the ellipsoid with the table. From the Lagrange equations the equation governing the motion of the ellipsoid and the constants of motion are obtained. The equations of motion are numerically solved for different initial conditions. Several simulation show realistic behaviour of the motion of the ellipsoid. Some of the rising effect is obtained from simulations, but unfortunately not the full effect. The reason for this is the lack of complexity of the model used.

To describe the full rising effect an improvement for the model is needed. This improvement could be achieved by implementing a more elaborate friction model. Despite the fact that the full rising effect does not follow from our model, a deeper understanding about the behaviour of the rolling and slipping ellipsoid is developed with the derivation and numerical solution of the equations of motion.

A Appendices

A.1 Runge-Kutta Python code

This is the Python code I wrote for solving the differential equations numerically using the fourth order Runge-Kutta method. In this case, as an example, that of the rolling ellipsoid on a table with the initial conditions and constants of motion from simulation 2 denoted in section 4.7.

```
import numpy as np

# This function returns the derivatives following from the
# differential equations.
def derivatives(state, t):
    #state = [theta,phi,psi,theta_deriv]
    theta = state[0]
    phi = state[1]
    psi = state[2]

    theta_deriv = state[3]

    xcm = state[4]
    ycm = state[5]
    zcm = state[6]

    h = np.sqrt(b**2*np.sin(theta)**2+a**2*np.cos(theta)**2)
    h_deriv = -((a**2-b**2)*np.sin(2*theta))/
            (2*np.sqrt(b**2*np.sin(theta)**2+a**2*np.cos(theta)**2))
    h_deriv_deriv = -((a**2-b**2)*
            (a**2*np.cos(theta)**4-b**2*np.sin(theta)**4))/
            ((b**2*np.sin(theta)**2+a**2*np.cos(theta)**2)**(3/2))

    f11 = I*np.sin(theta)**2 + m*(mu-1)**2*h_deriv*
            (np.sin(theta)**2*h_deriv - np.cos(theta)*np.sin(theta)*h)
    f12 = Is*np.cos(theta) + m*(mu-1)**2*h_deriv*
            (np.sin(theta)*h + np.cos(theta)*h_deriv)
    f21 = m*(mu-1)**2*(np.sin(theta)*h + np.cos(theta)*h_deriv)*
            (np.sin(theta)**2*h_deriv - np.cos(theta)*np.sin(theta)*h)
    f22 = Is + m*(mu-1)**2*(np.sin(theta)*h + np.cos(theta)*h_deriv)**2

    phi_deriv = (Lz*f22-Lz_p*f12)/(f22*f11-f21*f12)
    S = (Lz_p*f11-Lz*f21)/(f22*f11-f21*f12)
    psi_deriv = S-phi_deriv*np.cos(theta)

    vcmX = (mu-1)*((S-phi_deriv*np.cos(theta))*np.sin(theta)*h +
            (phi_deriv*np.sin(theta)**2+S*np.cos(theta))*h_deriv)
    vcmY = (mu-1)*theta_deriv*h
    vcmZ = theta_deriv*h_deriv
    vcmX_deriv = (mu-1)*((S - phi_deriv*np.cos(theta))*np.cos(theta)*h
            + (phi_deriv*np.sin(theta)**2+S*np.cos(theta))*h_deriv_deriv)

    vcmx = vcmX*np.cos(phi) - vcmY*np.sin(phi)
    vcmy = vcmX*np.sin(phi) + vcmY*np.cos(phi)
    vcmz = vcmZ

    first = -theta_deriv**2*m*(h*(mu-1)**2+h_deriv_deriv)*h_deriv
    second = I*phi_deriv**2*np.sin(theta)*np.cos(theta)
    third = -Is*S*phi_deriv*np.sin(theta)
    fourth = m*vcmX*vcmX_deriv
    fifth = -m*g*h_deriv

    theta_deriv_deriv = (first+second+third+fourth+fifth)/
            (I + m*h**2*(mu-1)**2 + m*h_deriv**2)

    return np.array([theta_deriv, phi_deriv, psi_deriv, theta_deriv_deriv,
            vcmx, vcmy, vcmz])

# This function returns the new state that follows from the old state
# using Runge-Kutta's method
def runge_kutta(state, t, dt, derivatives):
    k1 = derivatives(state,t)*dt
    k2 = derivatives(state + (0.5 * k1), t + 0.5 * dt)*dt
    k3 = derivatives(state + (0.5 * k2), t + 0.5 * dt)*dt
    k4 = derivatives(state + k3, t + dt)*dt

    state_new = state + (k1 + 2*k2 + 2*k3 + k4) / 6.

    return state_new

# Chosen and calculated values of used parameters
m = 65e-3 #kg
g = 9.81e3 #mm s^-2
a=31 #mm
b=22 #mm
d=0 #mm
x_cm = Moment_of_Inertia_3Degg.x_cm #mm
I = Moment_of_Inertia_3Degg.Iycm #mm^2 kg
Is = Moment_of_Inertia_3Degg.Izz #mm^2 kg

# Chosen constants of motion
Lz = 10 #mm^2 kg s^-1
Lz_p = 200 #mm^2 kg s^-1
mu=0.

# Chosen initial conditions, where state is given by:
```

```
# state = [theta,phi,psi,theta_deriv,xcm,ycm,zcm]
state0 = [np.pi/3,0,0,0,0,0,0]
state = np.array([state0])

# Time and time steps
dt=0.001
t_end = 1
t = np.arange(0,t_end,dt)

# For-loop in which from every previous state the new state is obtained
for i,time in enumerate(t[1:]):
    state_new = runge_kutta(state[i,:],time,dt,derivatives)
    state =  np.vstack((state, state_new))
```

A.2 Animation Python code

This is the Python code I wrote for making the rotating egg animations.

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import mpl_toolkits.mplot3d.axes3d as p3

# Functions for rotating around a certain axis
def rot_z(pos,alpha):
    x = pos[0]
    y = pos[1]
    z = pos[2]
    x_rx = np.cos(alpha)*x - np.sin(alpha)*y
    y_rx = np.sin(alpha)*x + np.cos(alpha)*y
    z_rx = z
    return (x_rx,y_rx,z_rx)

def rot_y(pos,beta):
    x = pos[0]
    y = pos[1]
    z = pos[2]
    x_rx = np.cos(beta)*x + np.sin(beta)*z
    y_rx = y
    z_rx = -np.sin(beta)*x + np.cos(beta)*z
    return (x_rx,y_rx,z_rx)

def rot_x(pos,gamma):
    x = pos[0]
    y = pos[1]
    z = pos[2]
    x_rx = x
    y_rx = np.cos(gamma)*y - np.sin(gamma)*z
    z_rx = np.sin(gamma)*y + np.cos(gamma)*z
    return (x_rx,y_rx,z_rx)


frn = len(t) # frame number of the animation

# Define a 3D matrix for the position and orientation of the egg in time
posx = np.zeros((len(x[0,:]), len(x[:,0]), frn))
posy = np.zeros((len(y[0,:]), len(y[:,0]), frn))
posz = np.zeros((len(z[0,:]), len(z[:,0]), frn))

# Euler angles and the position of the center of mass
# from the solution of the numerical method
theta = state[:,0]
phi = state[:,1]
psi = state[:,2]
xcm = state[:,4]
ycm = state[:,5]
zcm = state[:,6]

# For-loop trough time filling in the position and orientation of the egg
for i, hoeken in enumerate(np.transpose([psi,theta,phi])):
    psi = hoeken[0] #3-axis rotation
    theta = hoeken[1] #x'-axis rotation
    phi = hoeken[2] #z-axis rotation

    posx[:,:,i] = rot_z(rot_x(rot_z((x,y,z),psi),theta),phi)[0] + xcm[i]
    posy[:,:,i] = rot_z(rot_x(rot_z((x,y,z),psi),theta),phi)[1] + ycm[i]
    posz[:,:,i] = rot_z(rot_x(rot_z((x,y,z),psi),theta),phi)[2] + zcm[i]

# Function that sets the new animation frame
def update(i,posx,posy,posz,egg_animation):
    egg_animation[0].remove()
    egg_animation[0] = ax.plot_surface(posx[:,:,i],posy[:,:,i],
        posz[:,:,i],color="navajowhite")

# Figure in which the animation is plotted
fig = plt.figure(111)
ax = p3.Axes3D(fig)
egg_animation = [ax.plot_surface(x,y,z,color="navajowhite")]
ax.set_xlim3d(-80, 60)
ax.set_ylim3d(-80, 60)
ax.set_zlim3d(-50, 50)
ax.set_xlabel('$x$ (mm)')
ax.set_ylabel('$y$ (mm)')
ax.set_zlabel('$z$ (mm)')
ax.view_init(elev=30,azim=2)
```

```
# Animation is made
ani = animation.FuncAnimation(fig, update,
            fargs=(posx,posy,posz,egg_animation),frames=frn, interval=1,
            blit=False)

# Animation is saved
# fps=int(1/(3*dt)) means that time runs 3 times slower then normal
ani.save('3D_ellipsoid_test5.mp4',writer='ffmpeg',fps=int(1/(3*dt)))
```

References

M. S. Ashbaugh, C. C. Chiconc, and R. H. Cushman. The twisting tennis racket. 1991.

E. Cheever. Fourth order runge-kutta, 2005. URL `https://lpsa.swarthmore.edu/NumInt/NumIntFourth.html`.

Fowless and Cassiday. *Analytical Mechanics*. Brooks/Cole, Cengage Learning, 7 edition, 2005.

H. K. Moffatt and Y. Shimomura. Spinning eggs — a paradox resolved. 2002.