Mahdieh Nejati Javaremi
Final Project _ME314, Northwestern University, Fall Quarter 2014

1. Initial Proposal: Simulating the 3D rigid body rotation of a tennis racquet about its principal axis of inertia, in order to visualize the "Dzhanibekov effect":
    i. 6 degrees of freedom (3 linear {x, y, z} and 3 angular $\{\psi, \phi, \theta\}$)
    ii. The body has rotational inertia about three principal axes.
    iii. The rigid body is perturbed through impacting a surface
    iv. Rigid body modelled as a circular head and attached arm.
    v. No external forces.
    - Modifications:
        i. 3 degrees of freedom (3 angular $\{\psi, \phi, \theta\}$). This was done for the sake of computational efficiency. I started out with using 6 configuration variables, but by the time I wanted to include impacts, my computer would not be able to process it.
        ii. The rigid body is perturbed through impacting a point in 3D space.
        iii. The rigid body is modelled as a cuboid with l = xd, w = yd, and h = zd

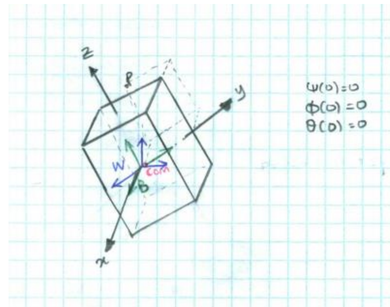


Figure 1 Cuboid body and space axes

2. The body frame and world frame share the same origin at the center of mass of the cuboid. The body frame is aligned with the principle axes of the cuboid, where the x-axis aligns with the principle axis of minimal inertia, the y-axes aligns with the principle axis of intermediate inertia, and the z-axis aligns with the principal axis of maximum inertia.
3.

The rigid body transformations were modelled as a sequence of rotations using Euler Angles.

$$\text{Rx} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi(t)) & -\sin(\psi(t)) \\ 0 & \sin(\psi(t)) & \cos(\psi(t)) \end{pmatrix}; \text{Ry} = \begin{pmatrix} \cos(\phi(t)) & 0 & \sin(\phi(t)) \\ 0 & 1 & 0 \\ -\sin(\phi(t)) & 0 & \cos(\phi(t)) \end{pmatrix}; \text{Rz} = \begin{pmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

And $g_{\text{wb}} =$

$$\begin{pmatrix} \cos(\theta(t))\cos(\phi(t)) & -\cos(\phi(t))\sin(\theta(t)) & \sin(\phi(t)) & 0 \\ \cos(\psi(t))\sin(\theta(t)) + \cos(\theta(t))\sin(\phi(t))\sin(\psi(t)) & \cos(\theta(t))\cos(\psi(t)) - \sin(\theta(t))\sin(\phi(t))\sin(\psi(t)) & -\cos(\phi(t))\sin(\psi(t)) & 0 \\ \sin(\theta(t))\sin(\psi(t)) - \cos(\theta(t))\cos(\psi(t))\sin(\phi(t)) & \cos(\psi(t))\sin(\theta(t))\sin(\phi(t)) + \cos(\theta(t))\sin(\psi(t)) & \cos(\phi(t))\cos(\psi(t)) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4. The system was modeled in the following way:
    - $\text{L} = \text{KE} = \frac{1}{2} * V_b{}^T . g_{\text{wb}} . V_b;$
    - Since the origin of the body frame and world frame coincide, the potential energy of the rigid body is essentially zero even though I have included the expression in my code as PE = $m * g * r_s[z]$ and L = KE-PE; where r_s is at the center of mass {0,0,0}
    - The Euler Lagrange equations are calculated as: $\frac{d}{dt}(\frac{\partial L}{\partial q'}) - \frac{\partial L}{\partial q} = 0$, where $q = \{\{\psi[t]\}, \{\phi[t]\}, \{\theta[t]\}\}$;
    - There are six impact conditions. Point pw is defined in the world frame:$\{\{-\text{xd}/2\}, \{\text{yd}/2 + \text{yd}/4\}, \{\text{zd}/2\}\}$. This point is transformed into the body frame $pb = g_{\text{bw}} . \underline{pw}$
        i. The six ϕ s are:
            1. ϕ1Impact=pb[[1]]-xd/2;
            2. ϕ2Impact=pb[[1]]+xd/2;
            3. ϕ3Impact= pb[[2]]-yd/2;
            4. ϕ4Impact= pb[[2]] +yd/2;
            5. ϕ5Impact=pb[[3]]-zd/2;
            6. ϕ6Impact= pb[[3]]+zd/2;

- The cuboid is constrained to rotate about its center of mass (origin of both frames). No translational motion is allowed.
- The impact update laws used were: $\frac{\partial L}{\partial q'}[\tau+] - \frac{\partial L}{\partial q'}[\tau-] = \lambda \frac{\partial \phi}{\partial q}$ and

$$[\frac{\partial L}{\partial q'}.q' - L(q,q')][\tau+] - [\frac{\partial L}{\partial q'}.q' - L(q,q')][\tau-] = 0$$

- Thirteen points of impact were calculated. The impact update laws used were:
  - i. Impact 1: occurs at t = 0.36556 seconds, satisfying: ϕ4Impact=0
  - ii. Impact 2: occurs at t = 1.17768 secs, satisfying: ϕ3Impact = 0
  - iii. Impact 3: occurs at t = 1.633 secs, satisfying: ϕ2Impact t=0
  - iv. Impact 4: occurs at t = 1.7523 secs, satisfying: ϕ3Impact = 0
  - v. Impact 5: occurs at t = 1.92646 secs, satisfying: ϕ2Impact=0
  - vi. Impact 6: occurs at t = 2.2036 secs, satisfying: ϕ2Impact=0
  - vii. Impact 7: occurs at t = 2.8557 sec, satisfying: ϕ1Impact=0
  - viii. Impact 8: occurs at t = 2.8583 sec, satisfying: ϕ2Impact=0
  - ix. Impact 9: occurs at t = 2.86 sec, satisfying: ϕ1Impact=0
  - x. Impact 10: occurs at t = 2.862 sec, satisfying: ϕ1Impact=0
  - xi. Impact 11: occurs at t = 2.862..secs, satisfying: ϕ6Impact=0
  - xii. Impact 12: occurs at t = 2.862..secs, satisfying: ϕ3Impact = 0

4. The code takes several seconds to load, and you will see warnings about Solve not being able to solve with inexact coefficients, also warnings regarding extrapolation being used instead of interpolations. But the code will run and you will see the animation. I hope you find the animations and the implementation of this theory as exciting as I have! There are six sections in my code. The first section is a set of rigid body functions I use to model the system. The second cell is my answers for the project. The third cell contains the Euler Lagrange Equations without impacts. I saved this in order to compare with the next three cells in my code that house Euler Equations.

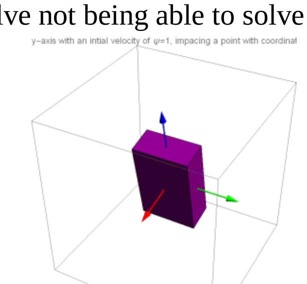


Figure 2before first impact.

The Dzhanibekov effect states that an object rotating about the intermediate axis of inertia is unstable. Therefore, even a small disturbance along other axes causes the object to ‘flip’. I was able to simulate this theory in Mathematica using the concepts learnt during the course.

The simulation starts with the cuboid rotating about its y- axis with an initial velocity of $\psi = 1$, shown in Figure 2. After the first impact, its trajectory changes to the right as seen in Figure 3. For the next few impacts, the cuboid is constantly hit and its orientation and angular velocities affected, unit at about the fifth and sixth impacts it becomes unstable and the axes suddenly change direction and the rotation is magnified over time. The cuboid quickly accelerates so the axes of rotation continually change. Four of the snapshots are shown in Table 1 below.

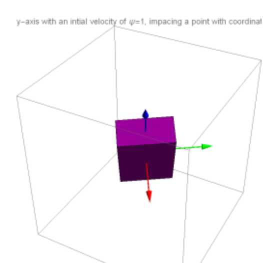
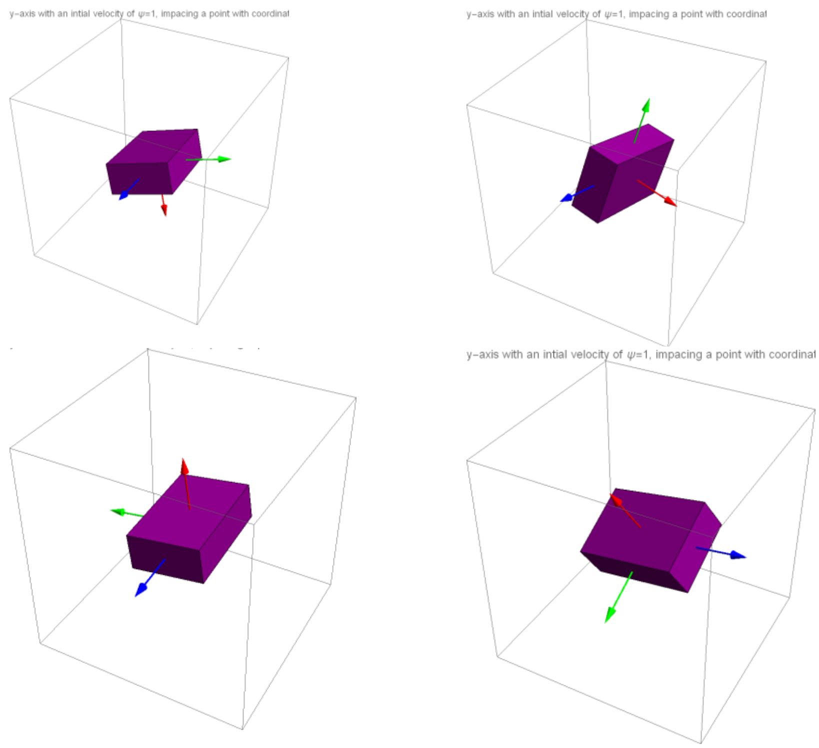


Figure 3after first impact. The angle of the cuboid is forced to rotate about its z axis as well.

Table 1 Unstable rotation (four frames, left to right, top to bottom).



The last few impacts happen extremely close together. At this point in the animation, you can see that the cuboid is angled such that as it responds to one impact, it hits the point again with its other side and continues to wobble back and forth. Also, at this point, the axis of rotation has changed from y to the z axis, and I think that this wobbling back and forth may be a result of the Dzhanibekov effect, where perturbations about min and max principle axes are stabilized. At this point, the axes are flipped. The animation show the results I was hoping to achieve.

Before adding any impacts, I investigated the equations without impacts using the Euler-Lagrange energy method. Additionally, I investigated the Euler equations where you can see the simulation of the block about each of its three principle axis (please refer to the last three code sections in my Mathematica notebook). If you give the cuboid a velocity of 1 or greater about either its minimum or maximum principle axes, and give a small perturbed velocity about the remaining two axes, there is no wobbling and the cuboid rotates stably as if there was zero velocity about the other two axes. However, using the same values, but with the greater velocity about the intermediate axis, the cube quickly becomes unstable until the direction of its axes changes 180 degrees. It was really interesting visualizing and simulating this curious physical property. However, the results from the Euler – Lagrange did not completely match those of the Euler equations. NDSolve tries to 'correct' the perturbations (roll, pitch, yaw singularities?) where the derivatives are not continuous. However, if the perturbations are increased in size, then we see the unstable rotations we expect. However, with the Euler Equations, NDSolve does not pose any issues. The solutions for $\{\psi[t]\}, \{\phi[t]\}, \{\theta[t]\}$ are shown below. I am using NDSolve to find solutions to both equations, however the Euler equations seem to give smoother trajectories, whereas the Euler Lagrange equations are stepped. I am guessing that the step size for NDSolve should be reduced for integrating the Euler Lagrange equations, or perhaps another integrator should be used. I find it curious that equations describing the same physical motion would have such discrepant outcomes using the same integrator. The trajectories of the three angles using Euler's equations are shown in Table 2, and in Table 3, the solutions from the Euler Lagrange energy method are shown.

Table 2 Euler Equations solutions

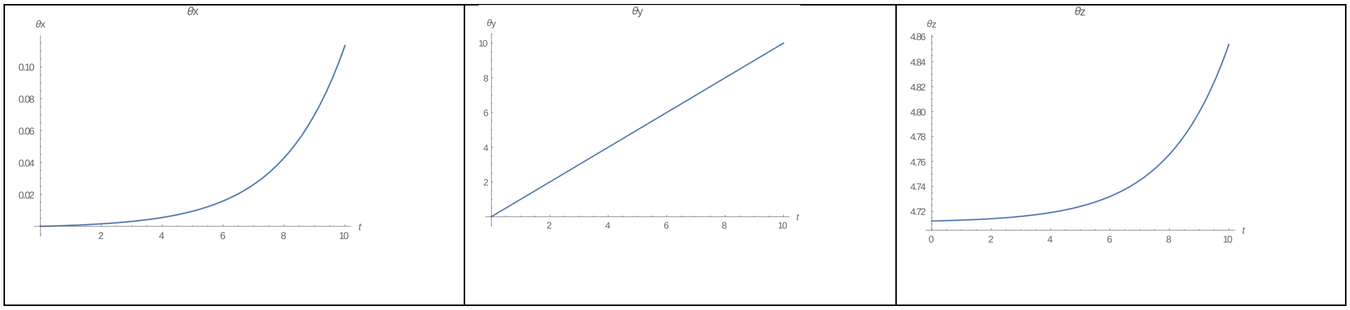
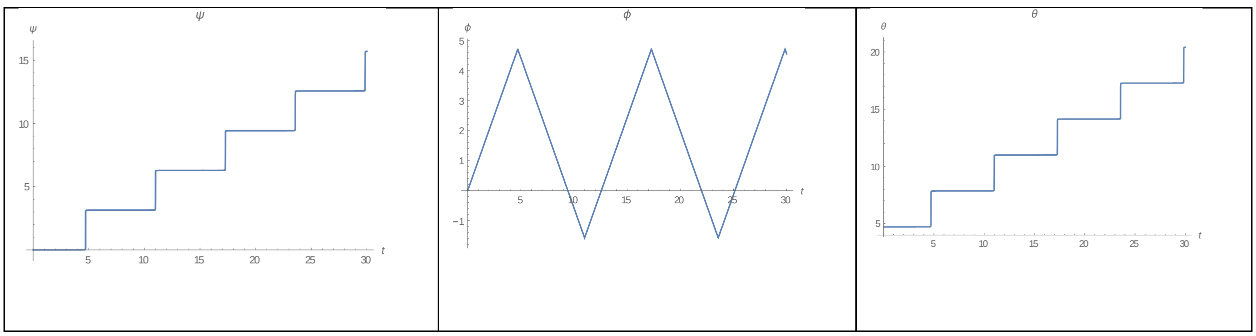


Table 3Euler Lagrange Equations



Plotting the generalized angular momentum, it is obvious that the Angular momentum constantly changes directions. In the absence of external torques, the angular momentum must be conserved. Though not it's direction, since the frame is non-inertial. In the non-inertial body frame, the direction of angular momentum changes. The rotational energy is also conserved.

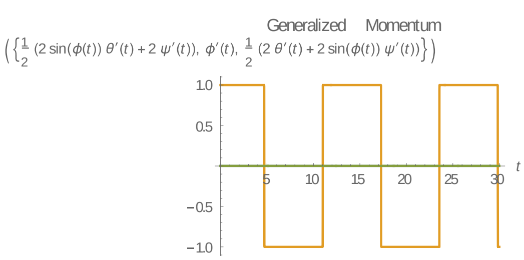


Figure 4 Angular Momentum

The fact that the impact times are extrapolated (estimated) rather than interpolated (sampled) there will be more error in my simulation. Also, decreasing the step length of NDSolve should help reduce the error of my simulation.