College of Engineering
School of Aerospace Engineering

AE6210 Advanced Dynamics I

Assignment

Final Project

Instructor:
Mayuresh Patil
Gtech AE Professor

Student:
Tomoki Koike
AE MS Student

April 27, 2022

Table of Contents

1 Problem Statement

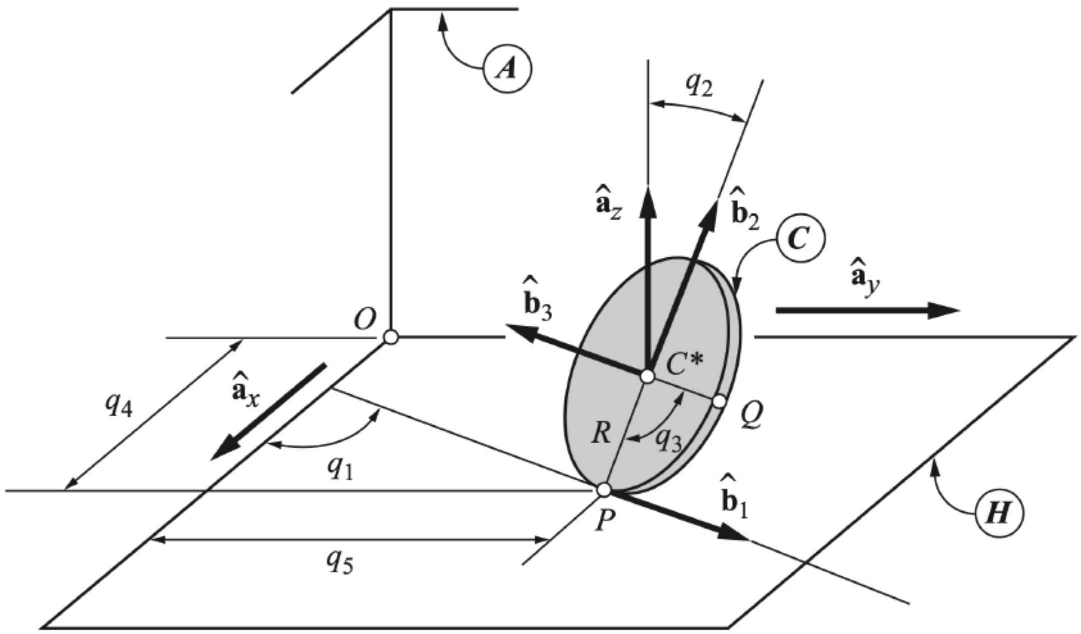

Figure 1: Kane's problem 2.7 from the book.

Assume that the disk and plane have a frictionless interface. The disk has 6 configuration variables, 1 holonomic constraint, 5 generalized coordinates, 0 non-holonomic constraints, and 5 generalized velocities. Derive the equations of motion using:

1) Kane's equations
2) Lagrange's equations
3) Newton-Euler equations

As you derive the equations you will have to:

a) Choose your generalized coordinates (you can choose the generalized coordinates that the book uses if you prefer or use any others).
b) Choose your motion variables/generalized velocities for Kane's approach (and if you wish for the Newton-Euler approach).

Show that all three approaches give equivalent equations of motion. You can do this either

1) Analytically: by deriving one set of equations from another.
2) Semi-analytically: by numerically calculating the accelerations using the different sets of equations for the same assumed state, i.e., assumed values for generalized coordinates and generalized velocities.
3) Computationally: by simulating the system using the different sets of equations for the same assumed initial condition.

2 Kinematics

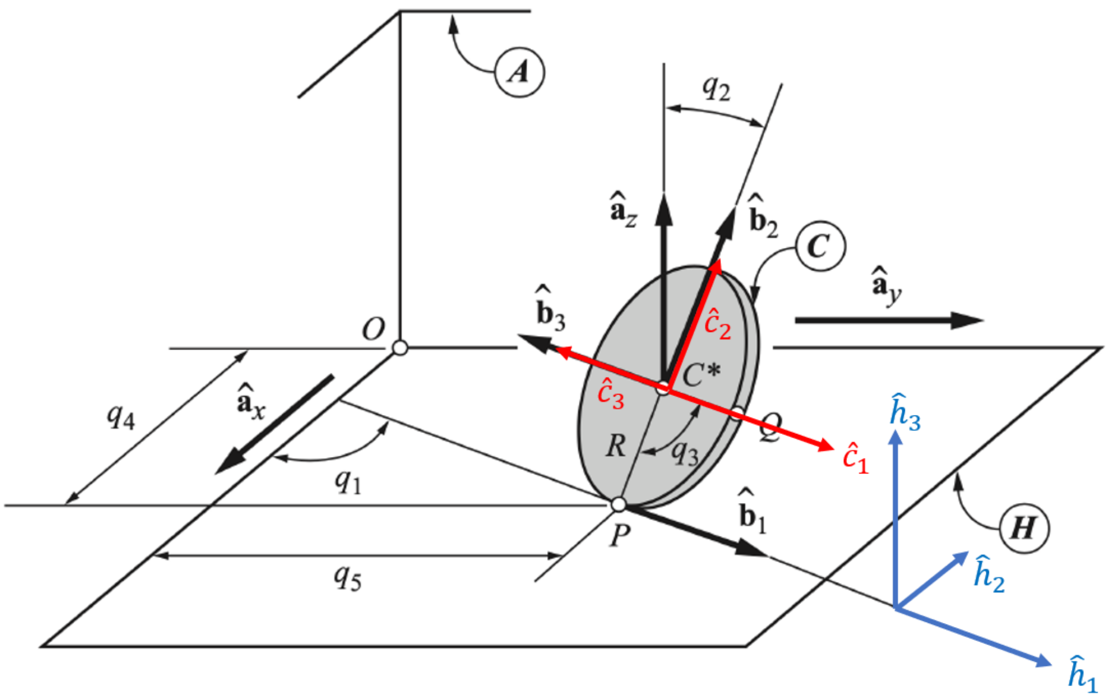

Figure 2: Problem diagram with coordinate frames.

Let the A-frame be $(\hat{a}_1, \hat{a}_2, \hat{a}_3) = (\hat{a}_x, \hat{a}_y, \hat{a}_z)$, and define new intermediate frames: C-frame and H-frame. Note that the body frame is the B-frame. For this problem, we define the generalized coordinates $(q_1, q_2, q_3, q_4, q_5)$ according to the diagram above. Once we do this we can define the relationship between each different frames with rotation matrices.

$$\hat{h} = R^H_A \hat{a}$$
$$\begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \end{bmatrix} = \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix}, \tag{2.1}$$

$$\hat{b} = R^B_H \hat{h}$$
$$\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & s_2 & c_2 \\ 0 & -c_2 & s_2 \end{bmatrix} \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \\ \hat{h}_3 \end{bmatrix}, \tag{2.2}$$

$$\hat{c} = R^C_B \hat{b}$$
$$\begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} = \begin{bmatrix} s_3 & -c_3 & 0 \\ c_3 & s_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix}. \tag{2.3}$$

The c and s indicate the simplified notation of the cosine and sine for the generalized coordinates q_1, q_2, q_3. From HW3 of this course we have derived the kinematic equations for these generalized coordinates. The angular velocity from the A-frame to the C-frame is expressed as

$$ {}^{A}\boldsymbol{\omega}^{C} = -\dot{q}_2\hat{\mathbf{b}}_1 + \dot{q}_1 c_2\hat{\mathbf{b}}_2 + (\dot{q}_1 s_2 + \dot{q}_3)\hat{\mathbf{b}}_3. \tag{2.4}$$

Thus, we choose the generalized velocities, $\mathbf{u}$ as

$$\begin{aligned} u_1 &= -\dot{q}_2 \\ u_2 &= \dot{q}_1 c_2 \\ u_3 &= \dot{q}_1 s_2 + \dot{q}_3. \end{aligned} \tag{2.5}$$

The angular velocity from the A-frame to B-frame can then be expressed using the generalized velocities to be

$$ {}^{A}\boldsymbol{\omega}^{B} = u_1\hat{\mathbf{b}}_1 + u_2\hat{\mathbf{b}}_2 + u_2 t_2\hat{\mathbf{b}}_3, \tag{2.6}$$

where t_2 is a shorthand notation for $\tan(q_2)$. Further the angular acceleration corresponding to (2.4) is

$$ {}^{A}\boldsymbol{\alpha}^{C} = \big(\dot{u}_1 + u_2(u_3 - u_2 t_2)\big)\hat{\mathbf{b}}_1 + \big(\dot{u}_2 - u_1(u_3 - u_2 t_2)\big)\hat{\mathbf{b}}_2 + \dot{u}_3\hat{\mathbf{b}}_3 = \alpha_1\hat{\mathbf{b}}_1 + \alpha_2\hat{\mathbf{b}}_2 + \alpha_3\hat{\mathbf{b}}_3. \tag{2.7}$$

Now, if we let the velocity of the geometric center, which is also the center of gravity point, be denoted as point $C^\star$ we can find the velocity of this in the A-frame using the generalized velocities v_1, v_2, v_3 and $\hat{\mathbf{b}}$ as follows.

$$ {}^{A}\mathbf{v}^{C^\star} = v_1\hat{\mathbf{b}}_1 + v_2\hat{\mathbf{b}}_2 + v_3\hat{\mathbf{b}}_3. \tag{2.8}$$

If we define two more generalized velocities $u_4 = \dot{q}_4$ and $u_5 = \dot{q}_5$ then obtain the expressions of

$$\begin{aligned} v_1 &= -Ru_2t_2 + u_4c_1 + u_5s_1 \\ v_2 &= -u_4s_1s_2 + u_5c_1s_2 \\ v_3 &= Ru_1 + u_4s_1c_2 - u_5c_1c_2 \end{aligned} \tag{2.9}$$

Subsequently the acceleration becomes

$$ {}^{A}\mathbf{a}^{C^\star} = a_1\hat{\mathbf{b}}_1 + a_2\hat{\mathbf{b}}_2 + a_3\hat{\mathbf{b}}_3, \tag{2.10}$$

where

$$\begin{aligned} a_1 &= -R\dot{u}_2t_2 + \dot{u}_4c_1 + \dot{u}_5s_1 + Ru_1u_2(2 + t_2^2) \\ a_2 &= -\dot{u}_4s_1s_2 + \dot{u}_5c_1s_2 - Ru_1^2 - Ru_2^2t_2^2 \\ a_3 &= R\dot{u}_1 + \dot{u}_4s_1c_2 - \dot{u}_5c_1c_2 + Ru_2^2t_2. \end{aligned} \tag{2.11}$$

Now for the next sections we define the following parameters: mass of M and moment of inertia of

$$\mathbf{I}^B = \frac{MR^2}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \tag{2.12}$$

3 Dynamics

3.1 Kane's Equations

Since we know the kinematics we begin with computing the partial angular velocities and partial velocities.

$$
{}^A\boldsymbol{\omega}_1^C = \frac{\partial}{\partial u_1}{}^A\boldsymbol{\omega}^C = \hat{\mathbf{b}}_1, \qquad {}^A\boldsymbol{\omega}_2^C = \frac{\partial}{\partial u_2}{}^A\boldsymbol{\omega}^C = \hat{\mathbf{b}}_2, \qquad {}^A\boldsymbol{\omega}_3^C = \frac{\partial}{\partial u_3}{}^A\boldsymbol{\omega}^C = \hat{\mathbf{b}}_3
$$

$$
{}^A\boldsymbol{\omega}_4^C = \frac{\partial}{\partial u_4}{}^A\boldsymbol{\omega}^C = 0, \qquad {}^A\boldsymbol{\omega}_5^C = \frac{\partial}{\partial u_5}{}^A\boldsymbol{\omega}^C = 0
$$

$$
{}^A\mathbf{v}_1^{C^\star} = R\hat{\mathbf{b}}_3, \qquad {}^A\mathbf{v}_2^{C^\star} = -Rt_2\hat{\mathbf{b}}_1, \qquad {}^A\mathbf{v}_3^{C^\star} = \vec{0}
$$

$$
{}^A\mathbf{v}_4^{C^\star} = c_1\hat{\mathbf{b}}_1 - s_1s_2\hat{\mathbf{b}}_2 + s_1c_2\hat{\mathbf{b}}_3, \qquad {}^A\mathbf{v}_5^{C^\star} = s_1\hat{\mathbf{b}}_1 + c_1s_2\hat{\mathbf{b}}_2 - c_1c_2\hat{\mathbf{b}}_3
$$

Now the inertia force and torque can be written as

$$
\mathbf{F}^\star = -M{}^A\mathbf{a}^{C^\star} = -M(a_1\hat{\mathbf{b}}_1 + a_2\hat{\mathbf{b}}_2 + a_3\hat{\mathbf{b}}_3), \tag{3.1}
$$

$$
\begin{aligned}
\mathbf{T}^\star &= -\mathbf{I}^B \cdot {}^A\boldsymbol{\alpha}^C - {}^A\boldsymbol{\omega}^B \times (\mathbf{I}^B \cdot {}^A\boldsymbol{\omega}^C) \\
&= \frac{R^2}{4}\left(\hat{\mathbf{b}}_1\hat{\mathbf{b}}_1 + \hat{\mathbf{b}}_2\hat{\mathbf{b}}_2 + 2\hat{\mathbf{b}}_3\hat{\mathbf{b}}_3\right) \cdot (\alpha_1\hat{\mathbf{b}}_1 + \alpha_2\hat{\mathbf{b}}_2 + \alpha_3\hat{\mathbf{b}}_3) \\
&\quad - (u_1\hat{\mathbf{b}}_1 + u_2\hat{\mathbf{b}}_2 + u_3\hat{\mathbf{b}}_3) \times \left[\frac{MR^2}{4}(\hat{\mathbf{b}}_1\hat{\mathbf{b}}_1 + \hat{\mathbf{b}}_2\hat{\mathbf{b}}_2 + 2\hat{\mathbf{b}}_3\hat{\mathbf{b}}_3) \cdot (u_1\hat{\mathbf{b}}_1 + u_2\hat{\mathbf{b}}_2 + u_3\hat{\mathbf{b}}_3)\right] \\
&= -\frac{MR^2}{4}(\dot{u}_1 + 3u_2u_3 - u_2^2t_2)\hat{\mathbf{b}}_1 - \frac{MR^2}{4}(\dot{u}_2 - 3u_1u_3 + 2u_1u_2t_2)\hat{\mathbf{b}}_2 - \frac{MR^2}{2}\dot{u}_3\hat{\mathbf{b}}_3
\end{aligned} \tag{3.2}
$$

Then the generalized inertia forces become as follows.

$$
\begin{aligned}
F_1^\star &= {}^A\boldsymbol{\omega}_1^C \cdot \mathbf{T}^\star + {}^A\mathbf{v}_1^{C^\star} \cdot \mathbf{F}^\star = -\frac{MR^2}{4}(\dot{u}_1 + 2u_2u_3 - u_2^2t_2) - MRa_3 \\
F_2^\star &= {}^A\boldsymbol{\omega}_2^C \cdot \mathbf{T}^\star + {}^A\mathbf{v}_2^{C^\star} \cdot \mathbf{F}^\star = -\frac{MR^2}{4}(\dot{u}_2 - 2u_1u_3 + u_1u_2t_2) + MRt_2a_1 \\
F_3^\star &= {}^A\boldsymbol{\omega}_3^C \cdot \mathbf{T}^\star + {}^A\mathbf{v}_3^{C^\star} \cdot \mathbf{F}^\star = -\frac{MR^2}{2}\dot{u}_3 \\
F_4^\star &= {}^A\boldsymbol{\omega}_4^C \cdot \mathbf{T}^\star + {}^A\mathbf{v}_4^{C^\star} \cdot \mathbf{F}^\star = -M(c_1a_1 - s_1s_2a_2 + s_1c_2a_3) \\
F_5^\star &= {}^A\boldsymbol{\omega}_5^C \cdot \mathbf{T}^\star + {}^A\mathbf{v}_5^{C^\star} \cdot \mathbf{F}^\star = -M(s_1a_1 + c_1s_2a_2 - c_1c_2a_3)
\end{aligned} \tag{3.3}
$$

The active force on this system is the gravity at the center of gravity of the disk

$$
\mathbf{R} = Mg(-\hat{\mathbf{h}}_3) = -Mg(c_2\hat{\mathbf{b}}_2 + s_2\hat{\mathbf{b}}_3). \tag{3.4}
$$

$$
F_1 = {}^A\mathbf{v}_1^{C^\star} \cdot \mathbf{R} = -MgRs_2, \qquad F_2 = {}^A\mathbf{v}_2^{C^\star} \cdot \mathbf{R} = 0, \qquad F_3 = {}^A\mathbf{v}_3^{C^\star} \cdot \mathbf{R} = 0 \tag{3.5}
$$

$$
F_4 = {}^A\mathbf{v}_4^{C^\star} \cdot \mathbf{R} = -Mg(s_1s_2c_2 - s_1s_2c_2) = 0, \qquad F_5 = {}^A\mathbf{v}_5^{C^\star} \cdot \mathbf{R} = -Mg(c_1c_2s_2 - c_1c_2s_2) = 0 \tag{3.6}
$$

Hence, the equations of motion is derived from the generalized active force and the generalized inertia force

$$
F_r + F_r^\star = 0. \tag{3.7}
$$

If we do the calculations for (3.7) and simplify things we end up with the following final expression. The derivation was done using MATLAB (refer to the code in Appendix 5.1).

$$
\begin{bmatrix}
5R & 0 & 0 & 4s_1c_2 & -4c_1c_2 \\
0 & R(1+4t_2^2) & 0 & -4c_1t_2 & -4s_1t_2 \\
0 & 0 & 1 & 0 & 0 \\
Rs_1c_2 & -Rc_1t_2 & 0 & 1 & 0 \\
Rc_1c_2 & Rs_1t_2 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \end{bmatrix}
=
\begin{bmatrix}
-Rt_2u_2^2 - 3Ru_2u_3 - 4gs_2 \\
2Rt_2(3+2t_2^2)u_1u_2 + 3Ru_1u_3 \\
0 \\
-Rs_1s_2u_1^2 - Rs_1s_2(1+t_2^2)u_2^2 - Rc_1(2+t_2^2)u_1u_2 \\
-Rc_1s_2u_1^2 - Rc_1s_2(1+t_2^2)u_2^2 + Rs_1(2+t_2^2)u_1u_2
\end{bmatrix}
$$

3.2 Lagrange's Equations

For the Lagrange method, we shall use the same configuration variables and the generalized velocities. For this system there is a potential energy and the kinetic energy consists of the translational and the rotational energy.

$$T = T_t + T_r = \frac{1}{2}M\left({}^{A}\mathbf{v}^{C^\star} \cdot {}^{A}\mathbf{v}^{C^\star}\right) + \frac{1}{2}({}^{A}\boldsymbol{\omega}^{C})^T\mathbf{I}^{B}\ {}^{A}\boldsymbol{\omega}^{C} \tag{3.1}$$

Then from (2.4),(2.5),(2.8), and (2.9) we have

$$T_t = \frac{M}{2}\left[\left(c_1\dot{q}_1 + s_1\dot{q}_5 - Rs_2\dot{q}_1\right)^2 + \left(Rq_2 + c_1c_2\dot{q}_5 - s_1c_2\dot{q}_4\right)^2 + s_2^2\left(s_1\dot{q}_4 - c_1\dot{q}_5\right)^2\right] \tag{3.2}$$

$$T_r = \frac{MR^2}{8}\left[c_2^2\dot{q}_1^2 + \dot{q}_2^2 + 2\left(s_2\dot{q}_1 + \dot{q}_3\right)^2\right]. \tag{3.3}$$

The potential energy is based on the location of the center of gravity of the disk tilting with the angle of q_2 which is

$$V = -MgR(1 - c_2). \tag{3.4}$$

Thus, the Lagrange equation becomes

$$L = T - V = T_t + T_r - V. \tag{3.5}$$

Now, if we apply the Euler-Lagrange equation with no external force on the body

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \qquad i = 1, 2, ..., 5 \tag{3.6}$$

and simplify the expression then we obtain the equations of motion (the computations were done using MATLAB and the code is in Appendix 5.2). The equations of motion can be written in the form of

$$\mathbf{E}\ddot{\mathbf{q}}_i = \mathbf{A}, \tag{3.7}$$

where $\mathbf{E} \in \mathbb{R}^{5\times5}$, $\mathbf{A} \in \mathbb{R}^{5\times1}$, and $\ddot{\mathbf{q}}_i \in \mathbb{R}^{5\times1}$, since for the double time derivative expression of the generalized coordinates we have a linear relationship.

$$\mathbf{E} = \begin{bmatrix} \frac{R^2}{2}(6 - 5c_2^2) & 0 & \frac{R}{2}s_2 & -c_1s_2 & -s_1s_2 \\ 0 & \frac{5R^2}{4} & 0 & -s_1c_2 & c_1c_2 \\ s_2 & 0 & 1 & 0 & 0 \\ Rc_1s_2 & Rc_2s_1 & 0 & -1 & 0 \\ Rs_1s_2 & -Rc_1c_2 & 0 & 0 & -1 \end{bmatrix} \tag{3.8}$$

and

$$\mathbf{A} = \begin{bmatrix} -\frac{5R}{2}c_2s_2\dot{q}_1\dot{q}_2 - \frac{R}{2}c_2\dot{q}_2\dot{q}_3 \\ \frac{5R}{4}s_2c_2\dot{q}_1^2 + \frac{R}{2}c_2\dot{q}_1\dot{q}_3 + gs_2 \\ -c_2\dot{q}_1\dot{q}_2 \\ Rs_1s_2\dot{q}_1^2 + Rs_1s_2\dot{q}_2^2 - 2Rc_1c_2\dot{q}_1\dot{q}_2 \\ -Rc_1c_2\dot{q}_1^2 - Rc_1s_2\dot{q}_2^2 - 2Rc_2s_1\dot{q}_1\dot{q}_2 \end{bmatrix} \tag{3.9}$$

3.3 Newton-Euler Equations

For the Newton-Euler we use the same configuration variables and generalized velocities. With the accelerations and angular accelerations derived in Section 2, we can compute the force and moment equality equations.

$$M {}^{A}\mathbf{a}^{C^\star} = \mathbf{F}$$
$$M(a_1\hat{\mathbf{b}}_1 + a_2\hat{\mathbf{b}}_2 + a_3\hat{\mathbf{b}}_3) = (N - Mg)(c_2\hat{\mathbf{b}}_2 + s_2\hat{\mathbf{b}}_3), \tag{3.1}$$

where N is the reaction force from the ground, and

$$\mathbf{I}^B \, {}^A\boldsymbol{\alpha}^C + {}^A\widetilde{\boldsymbol{\omega}}^B \mathbf{I}^B \, {}^A\boldsymbol{\omega}^C = -NRs_2\hat{\mathbf{b}}_1. \tag{3.2}$$

This gives six equations. However, we have only 5 unknowns so if we compute the unknowns using MATLAB we end up with the following equations (code in Appendix 5.3).

$$\begin{bmatrix} 0 & -MRt_2 & 0 & Mc_1 & Ms_1 & 0 \\ 0 & 0 & 0 & -Ms_1s_2 & Mc_1s_2 & -c_2 \\ MR & 0 & 0 & Ms_1c_2 & -Mc_1c_2 & -s_2 \\ MR^2 & 0 & 0 & 0 & 0 & 4Rs_2 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \\ N \end{bmatrix} = \begin{bmatrix} -MRu_1u_2(2+t_2^2) \\ MRu_1^2 + MRu_2^2t_2^2 - Mgc_2 \\ -Mu_2^2Rt_2 - Mgs_2 \\ MR^2t_2u_2^2 - MR^2u_2(u_3 - t_2u_2) - 2MR^2u_2u_3 \\ 3u_1u_3 - 2t_2u_1u_2 \\ 0 \end{bmatrix}$$

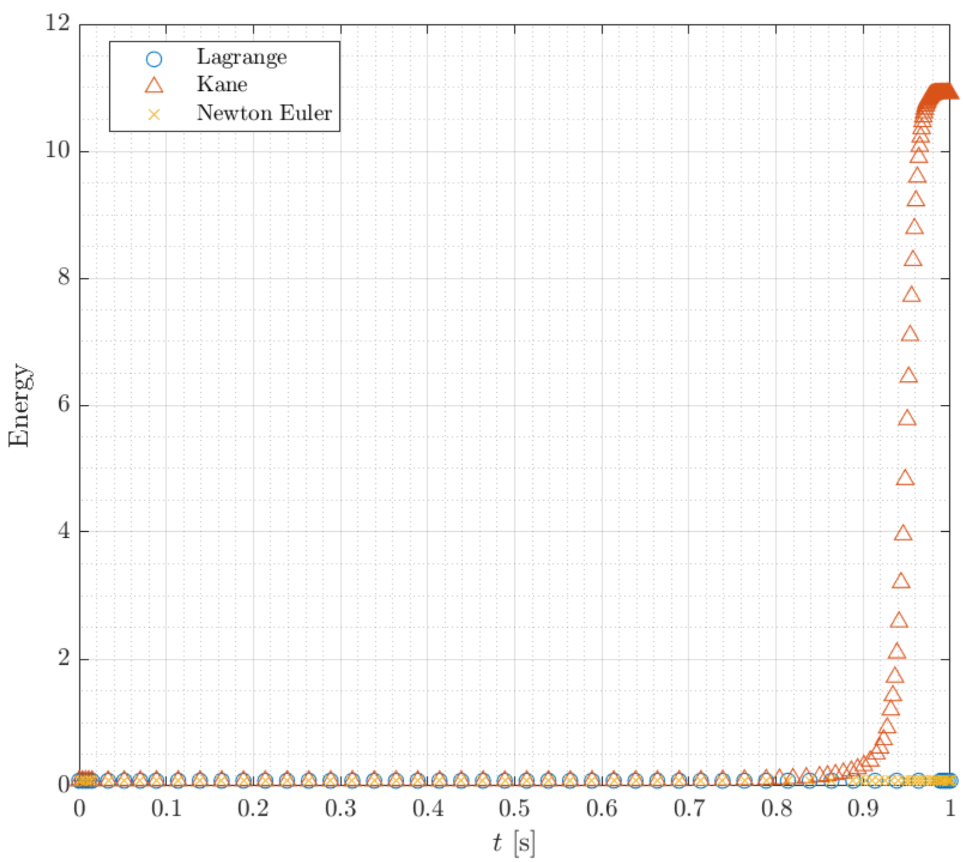

Figure 3: Energy conservation of the simulation results.

4 Simulation & Verification

For the simulation we use MATLAB's `ode45`. The necessary parameters are

$$M = 1, \quad R = 1, \quad g = 9.81. \tag{4.1}$$

The initial conditions are

$$q_{10} = 0.1745, \quad q_{20} = 0.0349, \quad q_{30} = 0.7854, \quad q_{40} = 5, \quad q_{50} = 1 \tag{4.2}$$
$$\dot{q}_{10} = 0.1, \quad \dot{q}_{20} = 0.01, \quad \dot{q}_{30} = 0.0203, \quad \dot{q}_{40} = 0.4, \quad \dot{q}_{50} = 0.1 \tag{4.3}$$

then

$$\dot{u}_{10} = 0.01, \quad \dot{u}_{20} = -0.1, \quad \dot{u}_{30} = 0.02, \quad \dot{u}_{40} = 0.4, \quad \dot{u}_{50} = 0.1 \tag{4.4}$$

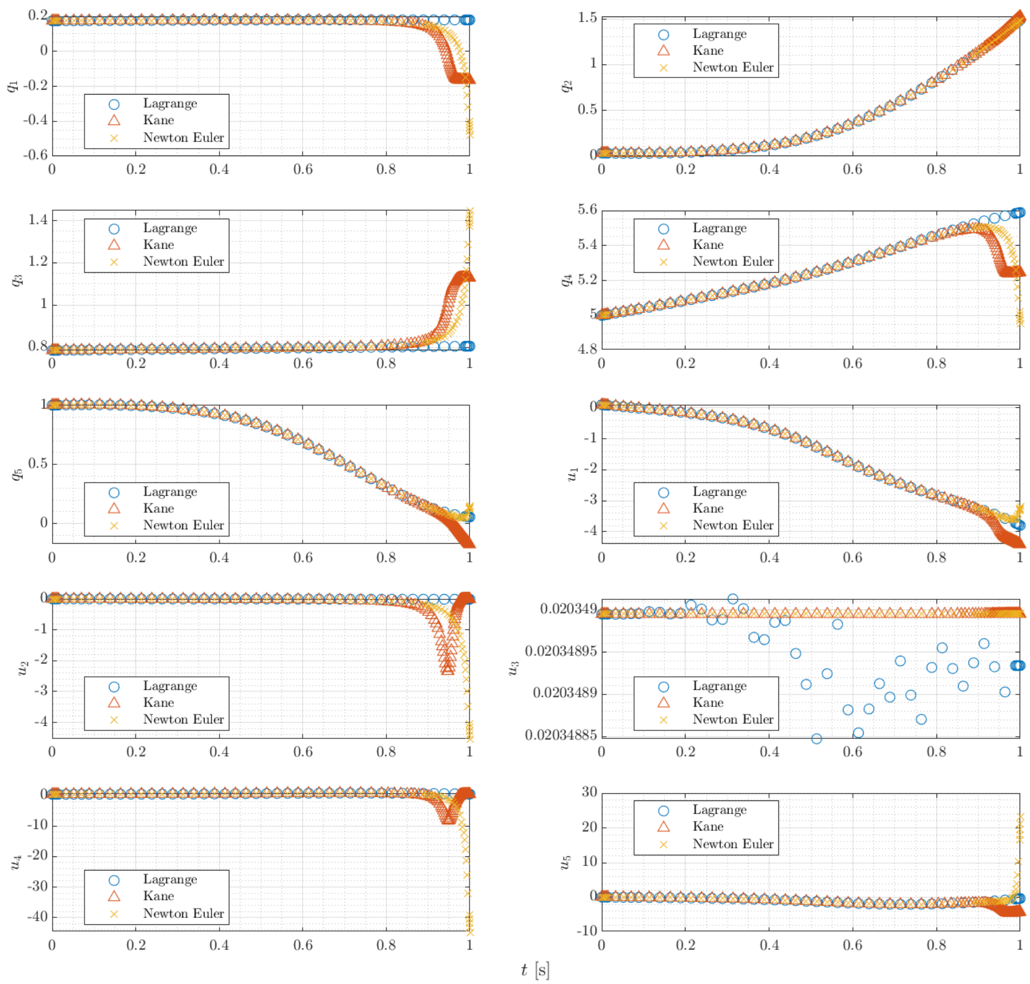

Figure 4: States for all three methods.

From figure 4 we can confirm that all 3 have an almost same trend. However, for the Kane's method we can see that at the end where the disk seems to tumble, i.e. q_2 becomes nearly 90 degrees, the matrix that we invert for the Kane's method becomes singular and the computations seem to lose its accuracy. Because of this singularity it was not possible to simulate any further than 1 second. The figure showing the conservation of energy suggests how this singularity effects the integration to fail since the energy is no longer conserved. The Lagrange and Newton-Euler seems to diverge as well if we simulate the 2 for a longer time span.

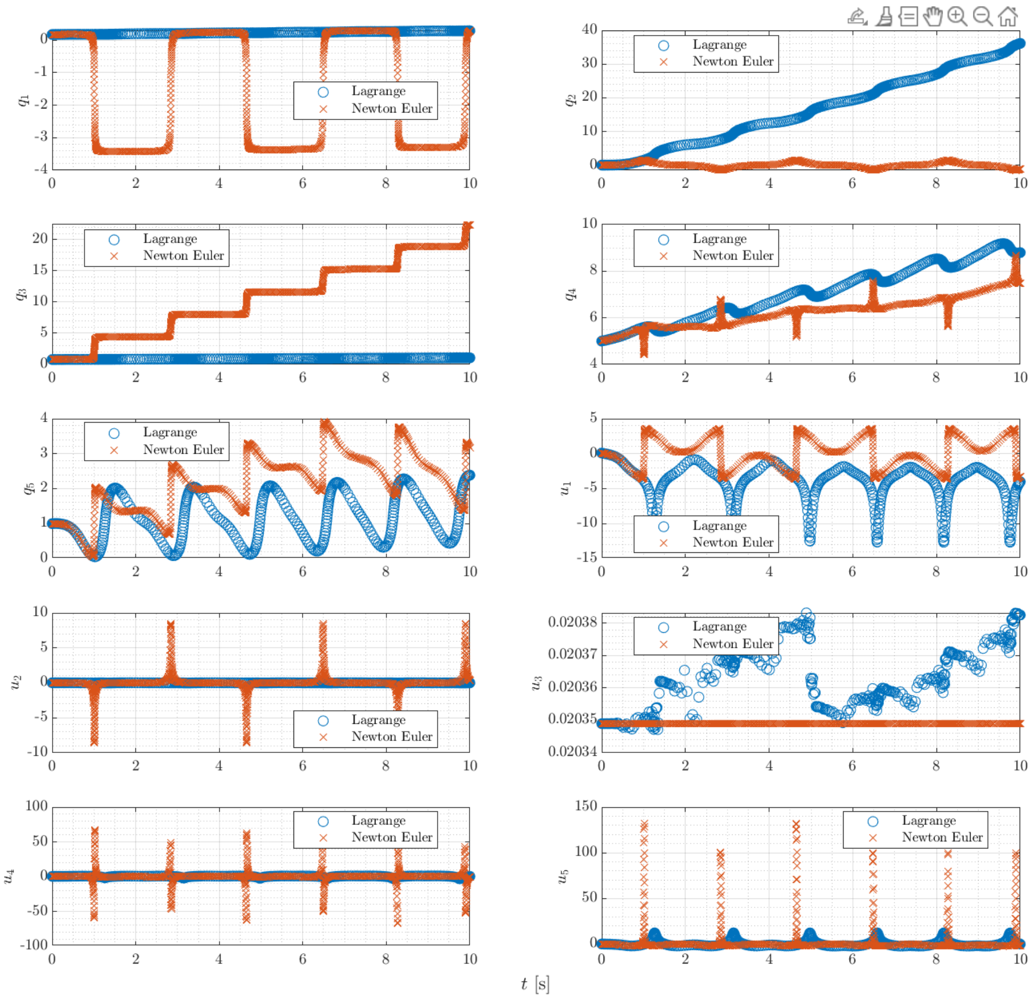

Figure 5: Simulation of Lagrange and Newton-Euler for 10 seconds.

5 Appendix

5.1 Kane's Method Derivation Code

```
%% Kanes

clear; close all; clc;
syms t real
syms q_1(t) q_2(t) q_3(t) q_4(t) q_5(t)
syms u_1(t) u_2(t) u_3(t) u_4(t) u_5(t)
syms M R g real

% Sine and cosines
s1 = sin(q_1); c1 = cos(q_1); s2 = sin(q_2); c2 = cos(q_2); t2 = tan(q_2);

% Derivatives
u1d = diff(u_1,t); u2d = diff(u_2,t); u3d = diff(u_3,t);
u4d = diff(u_4,t); u5d = diff(u_5,t);

% Acceleration
a1 = -R*u2d*t2 + u4d*c1 + u5d*s1 + R*u_1*u_2*(2 + t2^2);
a2 = -u4d*s1*s2 + u5d*c1*s2 - R*u_1^2 - R*u_2^2*t2^2;
a3 = R*u1d + u4d*s1*c2 - u5d*c1*c2 + R*u_2^2*t2;
aAC = [a1;a2;a3];

% Angular acceleration
alpha1 = u1d + u_2*(u_3 - u_2*t2);
alpha2 = u2d - u_1*(u_3 - u_2*t2);
alpha3 = u3d;
alphaAC = [alpha1;alpha2;alpha3];

% Angular velocity
wAC = [u_1;u_2;u_3];
% wAB = [u_1;u_2;u_2*t2];

% Partial velocities
v1 = [0; 0; R];
v2 = [-R*t2; 0; 0];
v3 = [0; 0; 0];
v4 = [c1; -s1*s2; s1*c2];
v5 = [s1; c1*s2; -c1*c2];

% Partial angular velocities
w1 = [1; 0; 0];
w2 = [0; 1; 0];
w3 = [0; 0; 1];
w4 = [0; 0; 0];
w5 = [0; 0; 0];

% Moment of inertia
I = M*R^2/4 * diag([1,1,2]);
%%
% Generalized active force
Rf = -M*g*[0; c2; s2];
```

```
F1 = (v1.') * Rf
F2 = (v2.') * Rf
F3 = (v3.') * Rf
F4 = (v4.') * Rf
F5 = (v5.') * Rf
%%
% Generalized inertia force
Tstar = collect(simplify(-I*alphaAC - cross(wAC,I*wAC)),M*R^2)
Fstar = simplify(-M*aAC)
Fstar1 = (w1.') * Tstar + (v1.') * Fstar
Fstar2 = (w2.') * Tstar + (v2.') * Fstar
Fstar3 = (w3.') * Tstar + (v3.') * Fstar
Fstar4 = (w4.') * Tstar + (v4.') * Fstar
Fstar5 = (w5.') * Tstar + (v5.') * Fstar
%%
% Equations of motion
eqn1 = simplify(expand(F1 + Fstar1))
eqn2 = simplify(expand(F2 + Fstar2))
eqn3 = simplify(expand(F3 + Fstar3))
eqn4 = simplify(expand(F4 + Fstar4))
eqn5 = simplify(expand(F5 + Fstar5))
%%
collect(eqn1,[u1d,u2d,u3d,u4d,u5d])
collect(eqn2,[u1d,u2d,u3d,u4d,u5d])
collect(eqn3,[u1d,u2d,u3d,u4d,u5d])
collect(eqn4,[u1d,u2d,u3d,u4d,u5d])
collect(eqn5,[u1d,u2d,u3d,u4d,u5d])
```

5.2 Lagrange's Method Derivation Code

```
%% Lagrange

syms t real
syms q_1(t) q_2(t) q_3(t) q_4(t) q_5(t)
syms M R g real

c1 = cos(q_1);
s1 = sin(q_1);
c2 = cos(q_2);
s2 = sin(q_2);
t2 = tan(q_2);

u1 = -diff(q_2,t);
u2 = diff(q_1,t)*c2;
u3 = diff(q_1,t)*s2+diff(q_3,t);
u4 = diff(q_4,t);
u5 = diff(q_5,t);

% Angular velocity
wAC = [u1;u2;u3];

v1 = -R*u2*t2+u4*c1+u5*s1;
% v2 = -u4*c1*s1+u5*c1*s2;
```

```
v2 = s2*(-u4*s1 + u5*c1);
v3 = R*u1+u4*s1*c2-u5*c1*c2;

% Velocity
vAC = [v1;v2;v3];

% Moment of inertia
Ib = M*R^2/4 * [1,0,0; 0,1,0; 0,0,2];
%%
% Kinetic and potential Energy
Tt = simplify(M*(vAC.')*vAC/2)  % translational
Tr = simplify((wAC.') * Ib * wAC / 2)  % rotational
V = -M*g*R*(1-c2);
%%
% Lagrange
L = simplify(Tt + Tr - V)
%%
% EOMs
vars = [diff(diff(q_1,t),t),...
        diff(diff(q_2,t),t),...
        diff(diff(q_3,t),t),...
        diff(diff(q_4,t),t),...
        diff(diff(q_5,t),t)];
eqn1 = simplify(diff(diff(L,diff(q_1,t)),t) - diff(L,q_1));
eqn2 = simplify(diff(diff(L,diff(q_2,t)),t) - diff(L,q_2));
eqn3 = simplify(diff(diff(L,diff(q_3,t)),t) - diff(L,q_3));
eqn4 = simplify(diff(diff(L,diff(q_4,t)),t) - diff(L,q_4));
eqn5 = simplify(diff(diff(L,diff(q_5,t)),t) - diff(L,q_5));
%%
collect(eqn1,vars)
collect(eqn2,vars)
collect(eqn3,vars)
collect(eqn4,vars)
collect(eqn5,vars)
```

5.3 Newton-Euler Method Derivation Code

```
%% Newton-Euler

clear; close all; clc;
syms t real
syms q_1(t) q_2(t) q_3(t) q_4(t) q_5(t)
syms u_1(t) u_2(t) u_3(t) u_4(t) u_5(t)
syms M R g N real

% Sine and cosines
s1 = sin(q_1); c1 = cos(q_1); s2 = sin(q_2); c2 = cos(q_2); t2 = tan(q_2);

% Derivatives
u1d = diff(u_1,t); u2d = diff(u_2,t); u3d = diff(u_3,t);
u4d = diff(u_4,t); u5d = diff(u_5,t);

% Acceleration
```

```
a1 = -R*u2d*t2 + u4d*c1 + u5d*s1 + R*u_1*u_2*(2 + t2^2);
a2 = -u4d*s1*s2 + u5d*c1*s2 - R*u_1^2 - R*u_2^2*t2^2;
a3 = R*u1d + u4d*s1*c2 - u5d*c1*c2 + R*u_2^2*t2;
aAC = [a1;a2;a3];

% Angular acceleration
alpha1 = u1d + u_2*(u_3 - u_2*t2);
alpha2 = u2d - u_1*(u_3 - u_2*t2);
alpha3 = u3d;
alphaAC = [alpha1;alpha2;alpha3];

% Angular velocity
wAC = [u_1;u_2;u_3];
wAB = [u_1;u_2;u_2*t2];
wAB = formula(wAB);
wABss = [0,-wAB(3),wAB(2); wAB(3),0,-wAB(1); -wAB(2),wAB(1),0];

% Moment of inertia
I = M*R^2/4 * diag([1,1,2]);
%%
eqn1 = M*a1;
eqn2 = M*a2; % - (N - M*g)*c2;
eqn3 = M*a3; % - (N - M*g)*s2;
%%
T = I*alphaAC + wABss*I*wAC
T = formula(T);
eqn4 = T(1) + N*R*s2;
eqn5 = T(2);
eqn6 = T(3);
%%
syms s_1 s_2 s_3 s_4 s_5
eqn1 = subs(eqn1,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
eqn2 = subs(eqn2,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
eqn3 = subs(eqn3,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
eqn4 = subs(eqn4,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
eqn5 = subs(eqn5,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
eqn6 = subs(eqn6,[u1d u2d u3d u4d u5d],[s_1 s_2 s_3 s_4 s_5]);
%%
clear
syms u_1 u_2 u_3 u_4 u_5 N real
syms c_1 s_1 c_2 s_2 t_2 real
syms M R g real
E = [0, -M*R*t_2, 0, M*c_1, M*s_1, 0;
     0, 0, 0, -M*s_1*s_2, M*c_1*s_2, -c_2;
     M*R, 0, 0, M*s_1*c_2, -M*c_1*c_2, -s_2;
     M*R^2, 0, 0, 0, 0, 4*R*s_2;
     0, 1, 0, 0, 0, 0;
     0, 0, 1, 0, 0, 0
    ];
A = [-M*R*u_1*u_2*(2+t_2^2);
     M*R*u_1^2 + M*R*u_2^2*t_2^2 - M*g*c_2;
     -M*u_2^2*R*t_2 - M*g*s_2;
     M*R^2*t_2*u_2^2 - M*R^2*u_2*(u_3 - t_2*u_2) - 2*M*R^2*u_2*u_3;
     3*u_1*u_3 - 2*t_2*u_1*u_2;
```

```
    0];
simplify(E \ A)
```

5.4 Kane's Method Function

```
function dzdt = disk_kane(t,z,R,g)
    % Preallocate output
    dzdt = zeros(10,1);

    % Unpack the variables
    q1 = z(1); q2 = z(2); q3 = z(3); q4 = z(4); q5 = z(5);
    u1 = z(6); u2 = z(7); u3 = z(8); u4 = z(9); u5 = z(10);

    % Trigs
    s1 = sin(q1); s2 = sin(q2); c1 = cos(q1); c2 = cos(q2); t2 = tan(q2);

    % Kinematic differential equation
    dzdt(1) = u2 * sec(q2);
    dzdt(2) = -u1;
    dzdt(3) = -u2*t2 + u3;
    dzdt(4) = u4;
    dzdt(5) = u5;

    % Dynamic differential equation
    % LHS matrix
    E = zeros(5,5);
    E(1,1) = 5*R;
    E(1,4) = 4*s1*c2;
    E(1,5) = -4*c1*c2;
    E(2,2) = R*(1 + 4*t2^2);
    E(2,4) = -4*c1*t2;
    E(2,5) = -4*s1*t2;
    E(3,3) = 1;
    E(4,1) = R*s1*c2;
    E(4,2) = -R*c1*t2;
    E(4,4) = 1;
    E(5,1) = R*c1*c2;
    E(5,2) = R*s1*t2;
    E(5,5) = -1;

    % RHS

    dzdt(6) = -R*t2*u2^2 - 3*R*u2*u3 - 4*g*s2;
    dzdt(7) = 2*R*t2*(3 + 2*t2^2)*u1*u2 + 3*R*u1*u3;

%     dzdt(6) = -3*R*t2*u2^2 - 2*R*u2*u3 - 4*g*s2;
%     dzdt(7) = R*t2*(7 + 4*t2^2)*u1*u2 + 2*R*u1*u3;

    dzdt(9) = -R*s1*s2*u1^2 - R*s1*s2*(1 + t2^2)*u2^2 - R*c1*(2 + t2^2)*u1*u2;
    dzdt(10) = -R*c1*s2*u1^2 - R*c1*s2*(1 + t2^2)*u2*2 + R*s1*(2 + t2^2)*u1*u2;

    % Output
    dzdt(6:10) = E \ dzdt(6:10);
```

```
end
```

5.5 Lagrange's Method Function

```
function dzdt = disk_lagrange(t,z,R,g)
    % Preallocate output
    dzdt = zeros(10,1);

    % Unpack the variables
    q1 = z(1); q2 = z(2); q3 = z(3); q4 = z(4); q5 = z(5);
    qd1 = z(6); qd2 = z(7); qd3 = z(8); qd4 = z(9); qd5 = z(10);

    % Trigs
    s1 = sin(q1); s2 = sin(q2); c1 = cos(q1); c2 = cos(q2); t2 = tan(q2);

    % Kinematic differential equation
    dzdt(1) = qd1;
    dzdt(2) = qd2;
    dzdt(3) = qd3;
    dzdt(4) = qd4;
    dzdt(5) = qd5;

    % Dynamic differential equations
    % LHS matrix
    E = zeros(5,5);
    E(1,1) = R^2/2*(6 - 5*c2^2);
    E(1,3) = R/2*s2;
    E(1,4) = -c1*s2;
    E(1,5) = -s1*s2;
    E(2,2) = 5*R^2/4;
    E(2,4) = -s1*c2;
    E(2,5) = c1*c2;
    E(3,1) = s2;
    E(3,3) = 1;
    E(4,1) = R*c1*s2;
    E(4,2) = R*c2*s1;
    E(4,4) = -1;
    E(5,1) = R*s1*s2;
    E(5,2) = -R*c1*c2;
    E(5,5) = -1;

    % RHS
    dzdt(6) = -5*R/2*c2*s2*qd1*qd2 - R/2*c2*qd2*qd3;
    dzdt(7) = 5*R/4*s2*c2*qd1^2 + R/2*c2*qd1*qd3 + g*s2;
    dzdt(8) = -c2*qd1*qd2;
    dzdt(9) = R*s1*s2*qd1^2 + R*s1*s2*qd2^2 - 2*R*c1*c2*qd1*qd2;
    dzdt(10) = -R*c1*c2*qd1^2 - R*c1*s2*qd2^2 - 2*R*c2*s1*qd1*qd2;

    % Output
    dzdt(6:10) = E \ dzdt(6:10);
end
```

5.6 Newton-Euler Method Function

```
function dzdt = disk_newtonEuler(t,z,R,g)
    % Preallocate output
    dzdt = zeros(10,1);

    % Unpack the variables
    q1 = z(1); q2 = z(2); q3 = z(3); q4 = z(4); q5 = z(5);
    u1 = z(6); u2 = z(7); u3 = z(8); u4 = z(9); u5 = z(10);

    % Trigs
    s1 = sin(q1); s2 = sin(q2); c1 = cos(q1); c2 = cos(q2); t2 = tan(q2);

    % Kinematic differential equation
    dzdt(1) = u2 * sec(q2);
    dzdt(2) = -u1;
    dzdt(3) = -u2*t2 + u3;
    dzdt(4) = u4;
    dzdt(5) = u5;

    % Dynamic differential equation
    den = c2^2 + 5*s2^2;

    dzdt(6) = (-4*g*c2^2*s2 + 2*R*c2^2*t2*u2^2 - 3*R*u3*c2^2*u2 + 4*R*c2*s2*t2^2*u2^2 ...
        + 4*R*c2*s2*u1^2 - 4*g*s2^3 - 2*R*s2^2*t2*u2^2 - 3*R*u3*s2^2*u2) / R / den;
    dzdt(7) = -2*t2*u1*u2 + 3*u1*u3;
    dzdt(9) = (-3*R*c1*c2^2*t2^2*u1*u2 + 3*R*c1*u3*c2^2*t2*u1 - 2*R*c1*c2^2*u1*u2 ...
        + 4*g*s1*c2*s2 - 3*R*s1*c2*t2*u2^2 + 3*R*s1*u3*c2*u2 - 15*R*c1*s2^2*t2^2*u1*u2 ...
        + 15*R*c1*u3*s2^2*t2*u1 - 10*R*c1*s2^2*u1*u2 - 5*R*s1*s2*t2^2*u2^2 - 5*R*s1*s2*u1^2)
             / den;
    dzdt(10) = (-3*R*s1*c2^2*t2^2*u1*u2 + 3*R*s1*u3*c2^2*t2*u1 - 2*R*s1*c2^2*u1*u2 ...
        - 4*g*c1*c2*s2 + 3*R*c1*c2*t2*u2^2 - 3*R*c1*u3*c2*u2 - 15*R*s1*s2^2*t2^2*u1*u2 ...
        + 15*R*s1*u3*s2^2*t2*u1 - 10*R*s1*s2^2*u1*u2 + 5*R*c1*s2*t2^2*u2^2 + 5*R*c1*s2*u1^2)
             / den;
end
```

5.7 Main Simulation

```
%% Simulation
% Tomoki Koike

%% Housekeeping commands
clear; close all; clc;
set(groot, 'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
warning('off');

%% Setup

M = 1;
R = 1;  % disk radius
g = 9.81;  % gravitational acceleration
```

```
I = M*R/4 * diag([1,1,2]);

% Initial conditions (Lagrange's)
q10 = deg2rad(10);
q20 = deg2rad(2);
q30 = deg2rad(45);
q40 = 5;
q50 = 1;

qd10 = 0.01;
qd20 = -0.1;
qd30 = 0.02;
qd40 = 0.4;
qd50 = 0.1;

IC.lagr = [q10;q20;q30;q40;q50;qd10;qd20;qd30;qd40;qd50];

% Initial conditions (Kane's)
u10 = -qd20;
u20 = qd10*cos(q20);
u30 = qd10*sin(q20) + qd30;
u40 = qd40;
u50 = qd50;

IC.kane = [q10;q20;q30;q40;q50;u10;u20;u30;u40;u50];

% Initial condition (Newton Euler)
IC.nwel = IC.kane;

%% Simulation

% Setup
% opts = odeset('RelTol',1e-3,'AbsTol',1e-5,Events=@diskTumbleEvents);  % tolerance
opts = odeset('RelTol',1e-3,'AbsTol',1e-5);  % tolerance
tspan = [0,1];

% Run
[t.kane,res.kane] = ode45(@(t,z) disk_kane(t,z,R,g),tspan,IC.kane,opts);  % Kane
[t.lagr,res.lagr] = ode45(@(t,z) disk_lagrange(t,z,R,g),tspan,IC.lagr,opts);  % Lagrange
[t.nwel,res.nwel] = ode45(@(t,z) disk_newtonEuler(t,z,R,g),tspan,IC.nwel,opts);  % Newton-
    Euler

% Convert lagrange q1dot q2dot and q3dot to u1 u2 u3
temp = res.lagr(:,6);
res.lagr(:,6) = -res.lagr(:,7);
res.lagr(:,7) = temp.*cos(res.lagr(:,2));
res.lagr(:,8) = temp.*sin(res.lagr(:,2)) + res.lagr(:,8);

%% Verification

% Energy
% Lagrange
q1 = res.lagr(:,1); q2 = res.lagr(:,2);
u1 = res.lagr(:,6); u2 = res.lagr(:,7); u3 = res.lagr(:,8);
```

```
u4 = res.lagr(:,9); u5 = res.lagr(:,10);
v1 = -R*u2.*tan(q2) + u4.*cos(q1) + u5.*sin(q1);
v2 = -u4.*sin(q1).*sin(q2) + u5.*cos(q1).*sin(q2);
v3 = R*u1 + u4.*sin(q1).*cos(q2) - u5.*cos(q1).*cos(q2);
J = zeros(length(t.lagr),1);
for i = 1:1:length(t.lagr)
    wAC = [u1(i);u2(i);u3(i)];
    J(i) = 0.5*M*(v1(i)^2 + v2(i)^2 + v3(i)^2) + 0.5*(wAC.')*I*wAC ...
        - M*g*R*(1 - cos(q2(i)));
end
energy.lagr = J;

% Newton-Euler
q1 = res.nwel(:,1); q2 = res.nwel(:,2);
u1 = res.nwel(:,6); u2 = res.nwel(:,7); u3 = res.nwel(:,8);
u4 = res.nwel(:,9); u5 = res.nwel(:,10);
v1 = -R*u2.*tan(q2) + u4.*cos(q1) + u5.*sin(q1);
v2 = -u4.*sin(q1).*sin(q2) + u5.*cos(q1).*sin(q2);
v3 = R*u1 + u4.*sin(q1).*cos(q2) - u5.*cos(q1).*cos(q2);
J = zeros(length(t.nwel),1);
for i = 1:length(t.nwel)
    wAC = [u1(i);u2(i);u3(i)];
    J(i) = 0.5*M*(v1(i)^2 + v2(i)^2 + v3(i)^2) + 0.5*(wAC.')*I*wAC ...
        - M*g*R*(1 - cos(q2(i)));
end
energy.nwel = J;

% Kane
q1 = res.kane(:,1); q2 = res.kane(:,2);
u1 = res.kane(:,6); u2 = res.kane(:,7); u3 = res.kane(:,8);
u4 = res.kane(:,9); u5 = res.kane(:,10);
v1 = -R*u2.*tan(q2) + u4.*cos(q1) + u5.*sin(q1);
v2 = -u4.*sin(q1).*sin(q2) + u5.*cos(q1).*sin(q2);
v3 = R*u1 + u4.*sin(q1).*cos(q2) - u5.*cos(q1).*cos(q2);
J = zeros(length(t.kane),1);
for i = 1:length(t.kane)
    wAC = [u1(i);u2(i);u3(i)];
    J(i) = 0.5*M*(v1(i)^2 + v2(i)^2 + v3(i)^2) + 0.5*(wAC.')*I*wAC ...
        - M*g*R*(1 - cos(q2(i)));
end
energy.kane = J;

fig = figure(Renderer="opengl",Position=[60 60 600 500]);
    plot(t.lagr,energy.lagr,'o',DisplayName='Lagrange')
    hold on; grid on; grid minor; box on;
    plot(t.kane,energy.kane,'^',DisplayName='Kane')
    plot(t.nwel,energy.nwel,'x',DisplayName='Newton Euler')
    hold off; legend(Location="best");
    xlabel('$t$ [s]')
    ylabel('Energy')
saveas(fig,"plots/energy.png")

%% Plot

```

```
ylabels = ["$q_1$", "$q_2$", "$q_3$", "$q_4$", "$q_5$",...
           "$u_1$", "$u_2$", "$u_3$", "$u_4$", "$u_5$"];

fig = figure(Renderer="opengl",Position=[60 60 1000 900]);
    for i = 1:10
        subplot(5,2,i)
        plot(t.lagr,res.lagr(:,i),'o',DisplayName="Lagrange")
        hold on; grid on; grid minor; box on;
        plot(t.kane,res.kane(:,i),'^',DisplayName="Kane")
        plot(t.nwel,res.nwel(:,i),'x',DisplayName="Newton Euler")
        hold off; legend(Location="best");
        ylabel(ylabels(i))
    end
    han=axes(fig,'visible','off');
    han.XLabel.Visible='on';
    xlabel(han,'$t$ [s]');
saveas(fig,"plots/3method_states.png");


%% Additional Function

function [value,isterminal,direction] = diskTumbleEvents(t,z)
    value = abs(z(2)) - pi/2;     % disk falls down
    isterminal = 1;   % Stop the integration
    direction = 0;   % Negative direction only
end
```