Soup to Simulation: The Skateboard Ramp

Modeling and Simulation

Instructions

A number of you identified in your feedback that the lectures were not working very well for you, and that you would find it helpful to see a more in-depth example. This exercise is an attempt to address both of these issues.

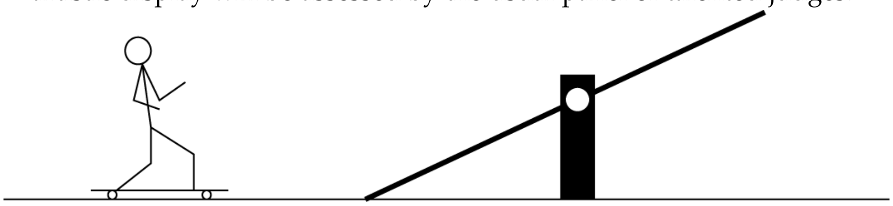
The exercise presents a set of questions to answer about a modeling a physical system: a "see-saw" skateboard ramp. You should attempt the questions on your own in studio; please feel free to talk to your colleagues and your instructor for ideas on how to proceed on each part.

If you're really stuck, *or* when you complete a section, talk to a ninja or an instructor – when you are stuck or or finished with a section, they will provide you with a *possible* solution for that section. Of course, other approaches are possible, so the solutions they provide exemplify one approach, which may not be the best approach (although it is might the simplest one, which makes it the best FIRST approach!).

If after reviewing the solutions for a given part you feel uncertain how the solution was obtained, or you feel that you could not do something similar on a somewhat simpler problem, **identify the questions and concerns you have**. There will be an opportunity on Wednesday to talk through questions about the solutions.

Announcement from World Skateboarding Federation

We are pleased to announce that the federation has just voted to include a new skateboard ramp competition in the next World Games. Unlike a typical skateboard ramp, this one is free to pivot about a support point. Skateboarders approach the ramp on a flat surface and then coast up the ramp; they are not allowed to put their feet down while on the ramp. If they go fast enough, the ramp will rotate and they will gracefully ride down the rotating ramp. Technical and artistic display will be assessed by the usual panel of talented judges.



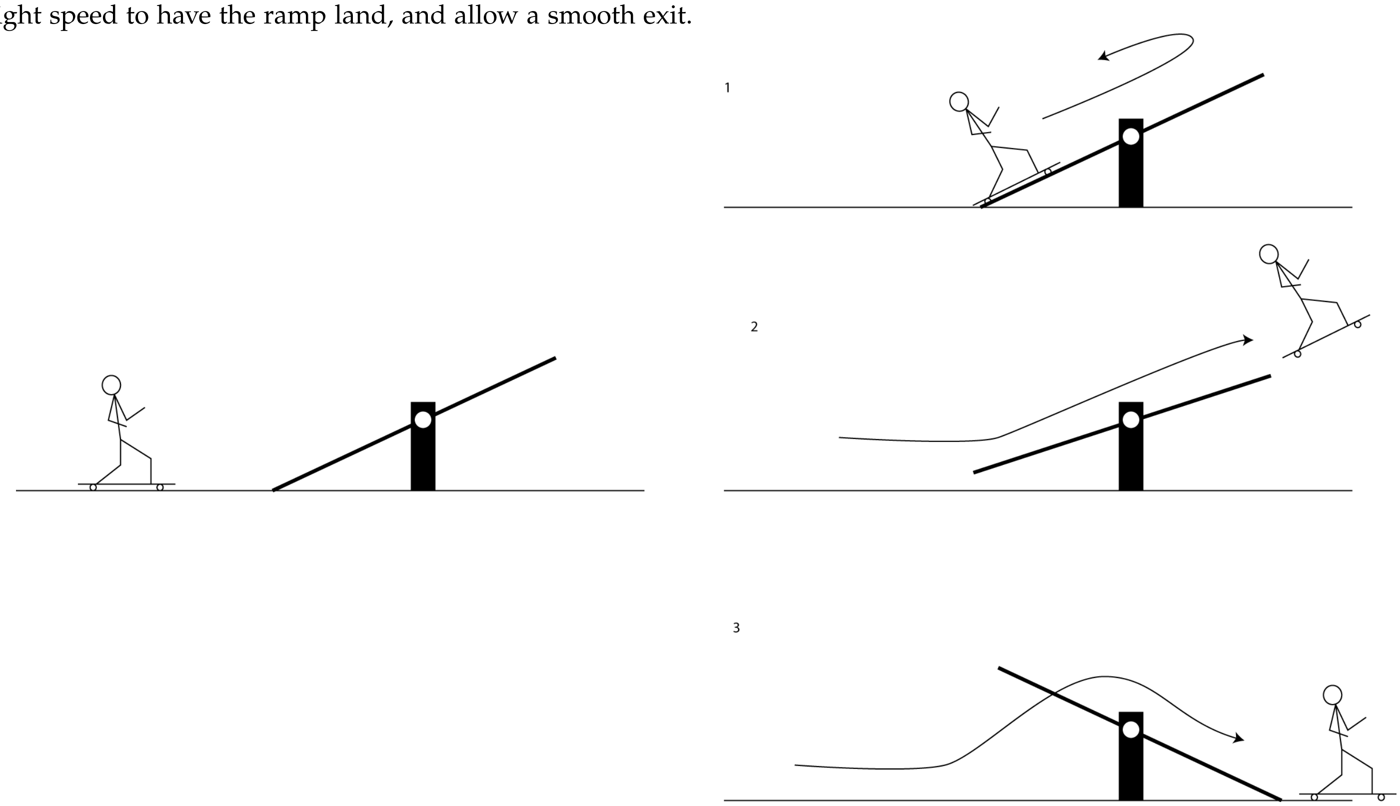
We are now looking for a team of engineers to design a new ramp which will allow a rider to accomplish the task outlined above. There are two design requirements:

- A simple model that allows an estimate of the length of ramp and mass of ramp for a typical rider approaching at a typical velocity.
- A dynamic simulation of the system, based on a model with enough detail to predict the minimum velocity a rider needs to get over the ramp and down the other side.

Key Frames

Create a set of key frames for this system that help you think through what the different parts of the system are, and how they move with respect to one another. Be sure to try to capture the different types of behavior you think might occur! These key frames should come in handy below both when you try to identify state variables, and when you try to do some estimation!

My best first guess at this is that there are three options for motion: first, the skateboarder could be going so slowly that they don't make it over. Second, the skateboarder could be so fast that they fly off the ramp. And finally, the skateboarder could be going at just the right speed to have the ramp land, and allow a smooth exit.



A comment is in order here: if we think through this, it seems pretty clear that in some situations, it might well be impossible for the system to work. For example, if the skateboarder weighed far less than the ramp, you could imagine that the skateboarder might either end up going off on the right in midair, or would slide back down and roll off on the left.

Abstraction Class, State Variables and Parameters

Given the system, the question, and your key frames, what class of abstraction do you expect to use for this system? In other words, how many different parts will you deal with? Will you be treating these parts as rigid bodies or as particles? Are the parts constrained in their motion, or unconstrained? Make a diagram and a table that define the state variables you will need to keep track of in order to model the dynamics of this system, as well as the relevant parameters for the system . Your table should include both a symbol for each state variable or parameter, a description of what the state variable or parameter tells you, and why it is important.

Looking at the scenarios outlined above, we can identify a number of different *phases* of motion we might want to model:

1. The time before the skateboarder goes onto the ramp. Presumably this is pretty simple – in fact, we might just want to START at the point when the skateboarder goes onto the ramp – or, even better, at the point when the skateboarder crosses the midpoint of the ramp!

2. The time when the skateboarder is on the ramp. This looks a lot more complex: the ramp can move (although it's limited by the ground to rotating through a particular angular range), and the skateboarder can move along the ramp.

3. The time after the skateboarder leaves the ramp (either by launching off the end of the ramp, or by neatly rolling off on to the ground on either the left or the right side). This is not so complex, as presumably we don't care how the ramp is moving once the skateboarder leaves the ramp. It's possible we'd want to think about the launch condition (keyframe 2 above) as distinct from the "clean" exit (keyframe 3), but given the question we want to answer, it might be enough just to know that the launch takes place.

So, even though there are multiple phases here, I propose that the one phase we really need to concentrate on is motion when the skateboarder is on the ramp.

During this phase, it's clear that the ramp and the skateboarder can EACH move – so I need at least 2 parts in this system.

The ramp's motion is clearly rotational, so I'll have to use a rigid body for that. Since it's fixed in the middle, this is a constrained rigid body.

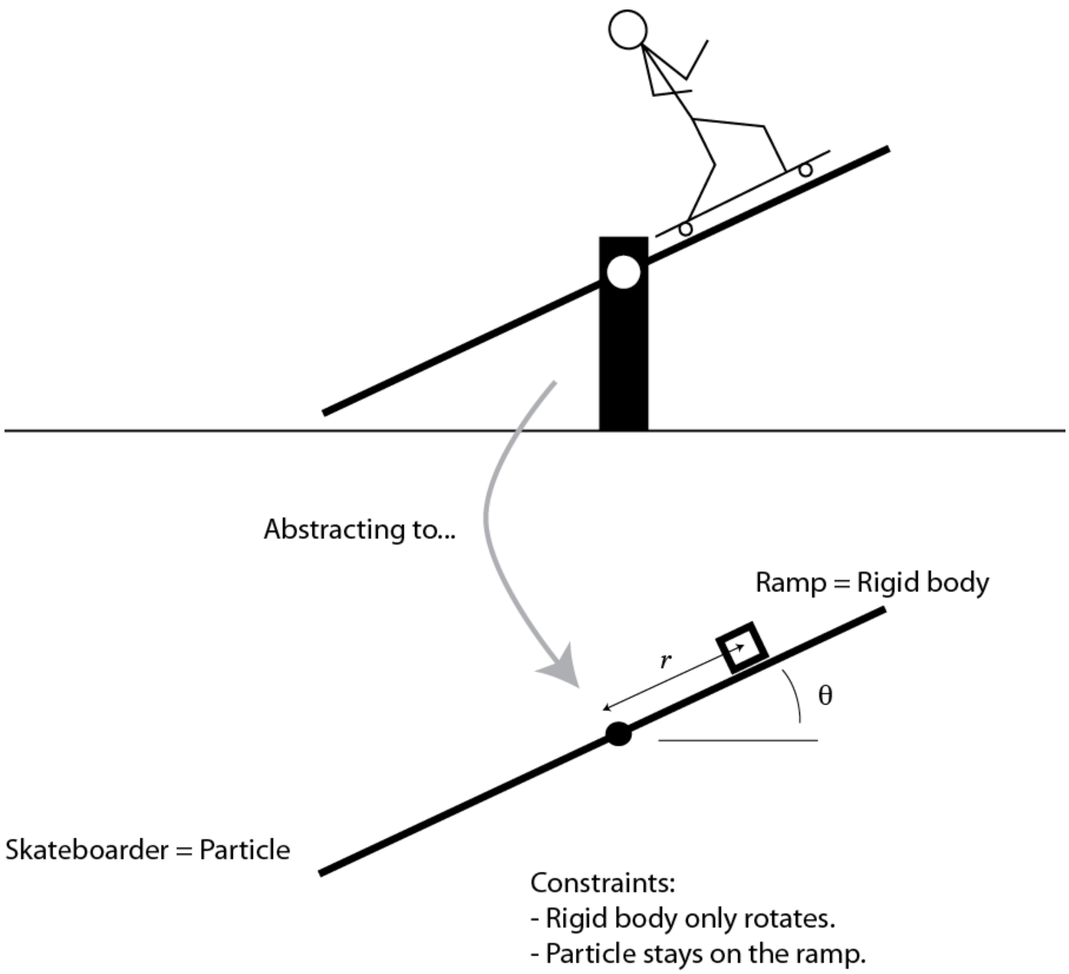
The skateboarder could be dealt with in any number of ways - from a particle (i.e., think about the skateboarder as a very compact "lump" that slides, with out without friction, along the ramp) to a rigid body that slides along the ramp and rotates, to a much more complex collection of rigid bodies (e.g., arms, legs, etc.) which are connected to each other in interesting ways (e.g., muscles and joints).

Since this is our first cut at the model, we should probably take the easiest approach that might give us some insight – i.e., treat the skateboarder as a particle.

Now, to the question of how many state variables there are: Clearly the position, and angular velocity of the ramp matter. Also, it's obvious that where the skateboarder *is* on the ramp matters, as does the skateboarder's speed. It appears that if you captured these four, you'd be in good shape.

From a parameters perspective, I would think that the mass of the skateboarder would matter, as would the mass and length of the ramp. Not sure what else...

So, my abstraction looks like this:



And my list of parameters and state variables is as follows:

State Variable/Parameter	Meaning, range, units, etc.	Why?
θ	Angular position of the ramp: runs from some positive angle to some negative angle, depending on ramp design. In radians.	Obvious by inspection.
ω	Angular velocity of the ramp: radians per second, could in principle be anything	Once the ramp is spinning, it's going to want to keep on spinning
r	Distance of rider from the pivot (positive = to the right of the pivot), in meters	The greater the distance, the greater the torque exerted
v_r	Time rate of change of r, in meters per second	Once the rider is moving, the rider wants to keep on moving.
m	Mass of the rider (kg)	Heavier riders will push down the ramp more.
L	Length of the ramp (m)	Determines both the initial angle and the length of time on the ramp, as well as the ramp's moment of inertia.
M	Mass of the ramp (in kg)	Determines the ramp's moment of inertia – bigger M means the ramp is harder to move

Now it's worth noting that the number of state variables is actually LOWER than you might guess it would be by just thinking about a particle and a rotating rigid body. For a particle, you generally need to know both its position ($\vec{r} = x\hat{i} + y\hat{j}$), and its momentum ($\vec{p} = m(\dot{x}\hat{i} + \dot{y}\hat{j})$, which gives you four state variables in two dimensions; for the rotating (but not translating) rigid body, you need to know its angular position (θ) and its angular momentum ($\vec{L} = I\omega\hat{k}$). However, since the sb is constrained to be on the surface of the ramp, x and y can be expressed purely in terms of how far along the ramp the sb is, and what the angle of the ramp is: $\vec{r} = r\cos\theta\hat{i} + r\sin\theta\hat{j}$. So, we end up needing two fewer state variables than we might expect otherwise.

Relevant Forces (and Torques)

Identify all of the different forces and torques that might be important in this system. Be sure to think about forces between parts of the system, as well as forces between the system and the rest of the universe. Make a table of the symbols, descriptions, and dependencies of the forces and torques. In your dependencies, identify both state variable and parameter dependencies!

A good way to think about this is to imagine "cutting" each part of the system out of the rest of the universe, and then thinking about what interactions you had to "cut" in order to accomplish this.

For the skateboarder, if we cut him/her out of the universe, we clearly have to cut the gravitational interaction with the earth (i.e., gravity exerts a force on the sb), and we also have to account for the fact that the ramp holds the sb up (i.e., there is a normal force interaction between the sb and the ramp). It's possible that we'd also want to account for a frictional interaction between the sb and the ramp, but since skateboards typically have pretty good bearings in their wheels, my first model will ignore this effect.

For the ramp, we've already identified a normal force interaction between the ramp and the sb – i.e., at any given instant, the ramp might be supplying momentum to the sb, or vice-versa. In addition to this, we observe that the ramp is pinned through its center of mass. This implies that WHATEVER other forces are applied, the pin will resist the force with an equal and opposite force so that the center of mass stays fixed. So really, we only need to think about the torques acting on the ramp – i.e., what are the forces being applied away from the COM. Since we've already decided not to include friction, the normal force appears to be the only one. We also could imagine there being a fricitonal torque at the pin – i.e., the pin might be rusty. But let's ignore this for now too.

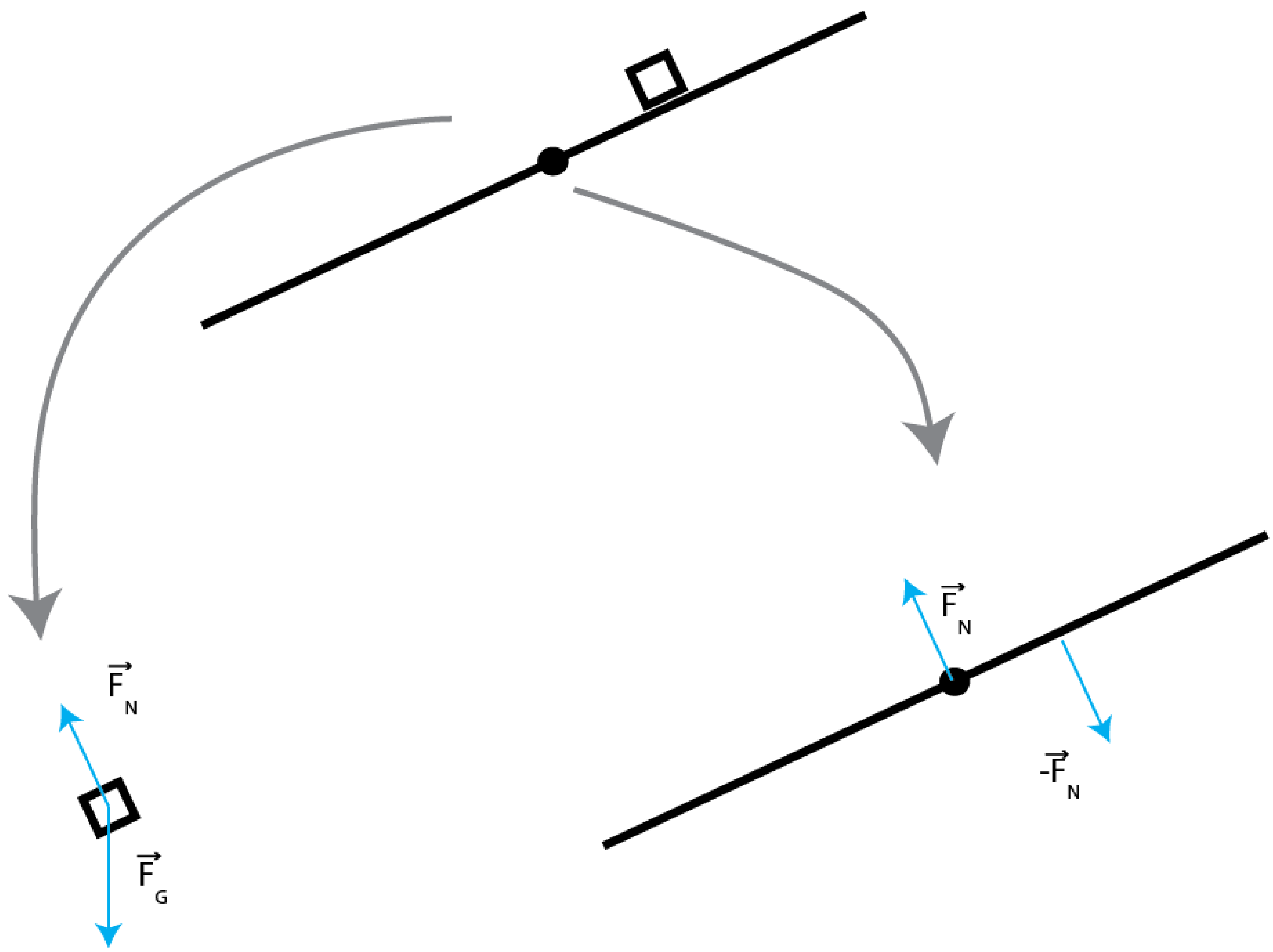
Thus, we have the following forces and torques that we think will be relevant:

Symbol	Meaning	Dependencies
$\vec{F}_G$	Gravitational force acting on the skateboarder	Depends on the sb mass; always points down.
$\vec{F}_N$	Normal force interaction between sb and ramp – constrains the sb not to fall through the ramp.	Always points normal to the ramp; pushes sb up, and ramp down. This is a constraint force, so it is whatever it needs to be given the current situation – i.e., it is likely to depend on pretty much everything!
$\vec{\tau}_N$	Torque exerted by normal force on ramp	This is a constraint torque, so it will depend on a lot of stuff. Clearly it directly depends on the position of the sb, and the normal force.

Graphical Abstraction: Free Body Diagrams

Draw free body diagrams for each particle or rigid body that you have identified as being a necessary component of your abstraction. Please use notation that is consistent with your force table!

Since we have two chunks here, we need two free body diagrams: one for the sb particle, and one for the ramp rigid body:



Note that for the rigid body, it is critical that we identify not only what the forces are, but where the forces are applied: we need to know the torque on the rigid body.

Simple Estimates

It is often a good idea to use your brain before you get really stuck into the start modeling. Can you set some bounds on this problem? Are there conditions under which it is obvious that the skateboarder would not make it over the ramp? Are there conditions under which it is obvious that the skateboarder will make it over?

Well, looking at the key frames, and thinking about this from an energy perspective, it seems pretty clear that a lower bound is set by the energy required to get up to the pivot. If the skateboarder's kinetic energy is below this, the outcome is guaranteed to be outcome 1 above.

Similarly, it seems pretty clear that if the skateboarder is going too fast, outcome 2 above will happen. One obvious example of this is if the skateboarder's kinetic energy is above the graviational potential energy of getting to the end of the (unrotated) ramp.

Thus, our first pass would at least give us a range of velocities that MIGHT work for the skateboarder:

$$mg\frac{L}{2}\sin\theta_0 < \frac{1}{2}mv^2 < mgL\sin\theta_0$$

where θ_0 is the ramp's designed angle, and L is the ramp's length.

We could put a further limit on this by thinking about the time it would take to make the ramp horizontal – i.e., to rotate the ramp through $\theta_0/2$ – by fixing the skateboarder at the far end of the ramp, and comparing this to the time for the skateboarder to roll back down to the middle of the ramp assuming the ramp was fixed. This would give us some insight into how the mass of the ramp would impact the problem – at the point when these two times are the same, you are around the "breakpoint" of the system; if $t_{horiz} \gg t_{L/2}$, you know the ramp is too heavy; if $t_{horiz} \ll t_{L/2}$, you can be pretty confident that the ramp is light enough. To do this, we'll assume constant accelerations and angular accelerations (this approach will turn this into much more of a "textbook" problem).

The moment of inertia of such a system would be given by

$$I = m(L/2)^2 + \frac{1}{12}ML^2$$

and, assuming that θ_0 is not too big, the gravitational torque would be *about*

$$\tau = -mgL/2$$

Note that this ignores the effect of angle in the torque, which is of course significant as θ gets larger. But so long as $\theta_0 < \frac{\pi}{8}$ or so, this should be an OK approximation. In this case, we'd expect that

$$\theta(t) = \theta_0 + 1/2\alpha t^2$$

where α is the angular momentum, given by

$$\alpha = \frac{\tau}{I}$$

This gives us a time to horizontal of

$$t_{horiz} \approx \sqrt{\frac{2\theta_0 I}{-\tau}} = \sqrt{\frac{2\theta_0(m(L/2)^2 + \frac{1}{12}ML^2)}{mgL/2}} = \sqrt{\frac{\theta_0 L(1 + \frac{M}{3m})}{g}}$$

We could compare this to the time that it would take the skateboarder to roll back a distance $L/2$:

$$t_{L/2} \approx \sqrt{\frac{2d}{a}} = \sqrt{\frac{L}{g \sin \theta_0}}$$

and since we've assumed θ_0 is relatively small, $\sin \theta_0 \approx \theta_0$ Thus, the condition that $t_{horiz} \approx t_{L/2}$ becomes

$$\theta_0 \sqrt{1 + \frac{M}{3m}} \approx 1$$

or, put another way, for a given value of M and m, the maximum value of θ_0 that will potentially work is about

$$\theta_{max} \approx \frac{1}{\sqrt{1 + \frac{M}{3m}}}$$

Note that this has the right limiting behavior: if $M \gg m$, $\theta_{max} \to 0$ – in other words, the ramp is too heavy for anything other than an almost horizontal initial angle. On the other hand, when $m \approx M$, θ_{max} can be quite a bit larger.

Now at this point we don't really have enough information to design the ramp, but we at least have some insight into what the bounds of the problem should be. In other words, we know that we probably want to be designing a ramp that has a height appropriate for typical skateboarder velocities – i.e.,

$$h_{pivot} < \frac{v_{max}^2}{2g}$$

and that the mass of ramp should be determined both by the mass of the skateboarder and by the initial angle of the ramp.

Equations of Motion

Using the principles of conservation of momentum and conservation of angular momentum, come up with second order differential equations that describe the time evolution of the system.

Since we're working in the world of "big, slow" objects, we're going to apply the central rules of Newtonian mechanics here: *momentum is conserved, and angular momentum is conserved.* Mathematically, this means that for ANY system (including parts of a bigger system),

$$\frac{d\vec{p}}{dt} = \sum \vec{F}$$

and

$$\frac{d\vec{L}}{dt} = \sum \vec{\tau}$$

where $\vec{p}$ is the *momentum* of the system, $\vec{L}$ is the *angular momentum* of the system, $\sum \vec{F}$ is the sum of all the forces being applied to the system, and $\sum \vec{\tau}$ is the sum of all the torques applied to the system.

So let's apply these rules to each part we're dealing with here: the skateboarder, and the ramp.

For the skateboarder, the momentum of the skateboarder is simply the skateboarder's mass times his or her velocity:

$$\vec{p} = m\vec{v}$$

Now, the velocity of the skateboarder requires a little thought. In general, the velocity is the time derivative of the position. The position is given by

$$\vec{r} = r\cos\theta\hat{\imath} + r\sin\theta\hat{\jmath}$$

where $\hat{\imath}$ is a unit vector in the x direction, and $\hat{\jmath}$ is a unit vector in the y direction. So, we can find the velocity by taking a time derivative of this. We of course need to remember the product rule here!

$$\vec{v} = \frac{d\vec{r}}{dt} = (\dot{r}\cos\theta - r\dot{\theta}\sin\theta)\hat{\imath} + (\dot{r}\sin\theta + r\dot{\theta}\cos\theta)\hat{\jmath}$$

Now it's pretty obvious, if you look at it, that things are going to get ugly here with all these sines and cosines. Let's introduce some new notation: $\hat{r}$, which indicates a unit vector pointing along the ramp, and $\hat{\theta}$, which indicates a unit vector normal to the ramp. If we think through the trig here, we get

$$\hat{r} = \cos\theta\hat{\imath} + \sin\theta\hat{\jmath}$$

and

$$\hat{\theta} = -\sin\theta\hat{\imath} + \cos\theta\hat{\jmath}$$

Furthermore, if we think about derivatives here,

$$\frac{d\hat{r}}{dt} = \dot{\theta}\hat{\theta}$$

and

$$\frac{d\hat{\theta}}{dt} = -\dot{\theta}\hat{r}$$

So then we can write much more neatly:

$$\vec{r} = r\hat{r}$$

and

$$\vec{v} = \frac{d\vec{r}}{dt} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta}$$

. Ah, isn't that prettier!

OK, so now we're ready to apply the law of momentum conservation to the skateboarder. If we expand the derivative of the momentum (using product rule again on the velocity expression above), we get

$$\frac{d\vec{p}}{dt} = \frac{d(m\vec{v})}{dt} = m(\ddot{r}\hat{r} + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r})$$

Now, the righthand side will be the sum of the forces applied to the skateboarder:

$$\sum \vec{F} = \vec{F}_G + \vec{F}_N$$

So how do we write these out? Well, $\vec{F}_G = -mg\hat{j}$, which is pretty simple. Unfortunately, we've written everything in terms of $\hat{\theta}$ and $\hat{r}$, so we need to write $\vec{F}_G$ this way too:

$$\vec{F}_G = -mg\cos\theta\hat{\theta} - mg\sin\theta\hat{r}$$

You'll have to draw yourself a few triangles to convince yourself of this.

$\vec{F}_N$ is a bit tougher. By inspection, it will only be in the $\hat{\theta}$ direction. It's tempting to say also that it has a magnitude that is simply $F_N = mg\cos\theta$. *But this would be very, very wrong!* Why? Well, the normal force is a constraint force – i.e., it is whatever it needs to be to keep the skateboarder from falling through the ramp. *If* the ramp were fixed, $mg\cos\theta$ would be right, because it would end up requiring that the net force was purely in the $\hat{r}$ direction. But since the ramp is not fixed, it is possible for the skateboarder to accelerate in the $\hat{\theta}$ direction too! So for now, we'll have to call F_N an unknown quantity.

Writing out the lefthand and righthand sides for the $\hat{r}$ components, we get the following differential equation:

$$m(\ddot{r} - r\dot{\theta}^2) = -mg\sin\theta$$

and for the $\hat{\theta}$ components, we obtain

$$m(r\ddot{\theta} + 2\dot{r}\dot{\theta}) = -mg\cos\theta + F_N$$

If we pause and look at these, in some ways you could say we have two equations and three unknowns: $\ddot{r}$, $th\ddot{e}ta$, and F_N. To get our third equation, we need to look at the other chunk in the system: the ramp.

Now, the ramp is a rigid body, which is constrained only to rotate. This means that the only interesting equation will be the angular momentum equation: the linear momentum of the ramp is ALWAYS zero. The angular momentum, on the other hand, will only contain a rotational component:

$$\vec{L} = I\dot{\theta}\hat{k}$$

where I is the moment of inertia of the ramp about its pivot, and $\hat{k}$ is a unit vector pointing out of the page (angular momentum points along the axis of rotation).

Calculating the derivative of this is pretty simple, since both I and $\hat{k}$ don't change in time:

$$\frac{d\vec{L}}{dt} = I\ddot{\theta}\hat{k}$$

Now, to calculate the RHS of the angular momentum conservation equation, we need to figure out the torque on the ramp. This is pretty easy too, as the only force that is being applied away from the pivot is the negative of the normal force. This is being applied perpendicular to the moment arm, so by inspection,

$$\sum \vec{\tau} = \vec{r} \times -\vec{F}_N = -rF_N\hat{k}$$

where $\vec{r}$ is the location of the skateboarder where the normal force is being applied.

Thus our angular momentum conservation equation becomes

$$I\ddot{\theta} = -rF_N$$

Now we're good to go: we have three equations, and three things we want to solve for. Manipulating algebraically, we obtain:

$$F_N = -\frac{I\ddot{\theta}}{r}$$

and

$$\ddot{\theta} = \frac{-mgr\cos\theta - 2mr\dot{r}\dot{\theta}}{mr^2 + I}$$

and finally

$$\ddot{r} = -g\sin\theta + r\dot{\theta}^2$$

Whew!

Of course, these equations only hold if the ramp free to move. Let's think about the other condition... We'll say that the ramp can rotate between θ_{min} and θ_{max}. Then, if θ drops below θ_{min}, then a couple of thing happen. First, it seems pretty clear that $\dot{\theta}$ would have to go to zero (i.e., the ramp would stop moving), and it also seems clear that the only way the ramp will start moving again is if $\ddot{\theta} > 0$ – i.e., if the skateboarder is in a location that leads to the ramp getting lifted. A similar situation applies when θ gets up to θ_{max} (although, of course, the condition for the ramp to move here is that $\ddot{\theta} < 0$).

Thinking it through

Whenever you have some complicated governing equations, it's a good idea to try to explain why they make sense. To the greatest extent possible, explain each term in your DE's.

Let's look first at the equation for $\ddot{r}$. The first term on the RHS side, $-g \sin \theta$, is a classic "block on a ramp" term: if the ramp is slanted, the block tends to accelerate down the ramp. The second term, $r\dot{\theta}^2$, is relatively easy to make sense of as well – if the ramp is spinning at some velocity, the block's speed along the ramp will increase in an outward direction. For example, imagine a bead on a spinning wire: the bead will get flung off the end of the wire. This looks like a so-called "centrifugal force" (which of course is not a real force, but a result of the fact that the block would have to accelerate in order to stay at a given r on the ramp!).

Now, if we examine $\ddot{\theta}$, there are three things that require interpretation: the two terms in the numerator, and the denominator. The first term in the numerator, $-mgr \cos \theta$, looks like the traditional torque due to a gravitational normal force – pretty easy to buy. The second term,$-2mr\dot{r}\dot{\theta}$, takes some more thought: it says that if the block is moving along the ramp, and the ramp is rotating, there is an additional torque that tends to slow down the ramp. This takes some thought, but makes sense if you think about it. Finally, the denominator, $mr^2 + I$, is pretty easy to interpret: it's the total moment of inertia of the system (ramp + skateboarder).

From second order to first order

Once you have your differential equations, you'll want to transform them into a set of first order DE's that will be compatible with an ODE solver. Do so!

This should be pretty straightforward: we just need to do some relabeling here.

As noted in our state variables, we have θ and ω, and r and v_r. Clearly two of our DE's are the simple definitional ones:

$$\frac{dr}{dt} = v_r$$

$$\frac{d\theta}{dt} = \omega$$

We obtain our other two DE's for ω and v_r by substituting these in the equation above:

$$\frac{d\omega}{dt} = \frac{-mgr\cos\theta - 2mrv_r\omega}{mr^2 + I}$$

and finally

$$\frac{dv_r}{dt} = -g\sin\theta + r\omega^2$$

Implement

Create the core functions for building a simulation of the system – in other words, write an appropriate "slope" function that can be used with `ode45`*.*

Note that once you've done a lot of work on paper, creating the simulation is pretty straightforward. Here's what I came up with as a first pass:

```
function res = skateboard_slope_func(t,W)
        % unpack
        R = W(1:2);  % unpack r and theta
        V = W(3:4);  % unpack v_r and omega

        % define parameters
        m = 70 % mass of skateboarder, kg
        MR= 70  % mass of ramp, kg
        L = 5    % total length of ramp, meters
        g = 10 % meters per second^2

        % calculate the slopes
        dRdt = V; % The simple first order DEs
        dVdt = acceleration_func(t,W); % The complex first order DE's we calculated

        % pass out the results
        res = [dRdt;dVdt];

    function res = acceleration_func(t,W)
    % Calculate second derivative of r and second derivative of theta
         r = W(1);
         theta = W(2);
         rdot = W(3);
         thdot = W(4);
         rdotdot = -g*sin(theta) + r*thdot^2;
         thdotdot = (-m*g*r*cos(theta) - 2*m*r*rdot*thdot)/(m*r^2 + 1/12*MR*L^2);
         res = [rdotdot;thdotdot];
    end

end
```

Now, there is still a LOT of work to be done here. This does not have any of the event catching you might need around the ramp hitting the ground or around the skateboarder launching off the end of the ramp. But this code at least could form a core...