Simulating Simple Roller Coaster Physics for Animation and Interactive Applets

Michael R. Gallis
Penn State Schuylkill
mrg3@psu.edu

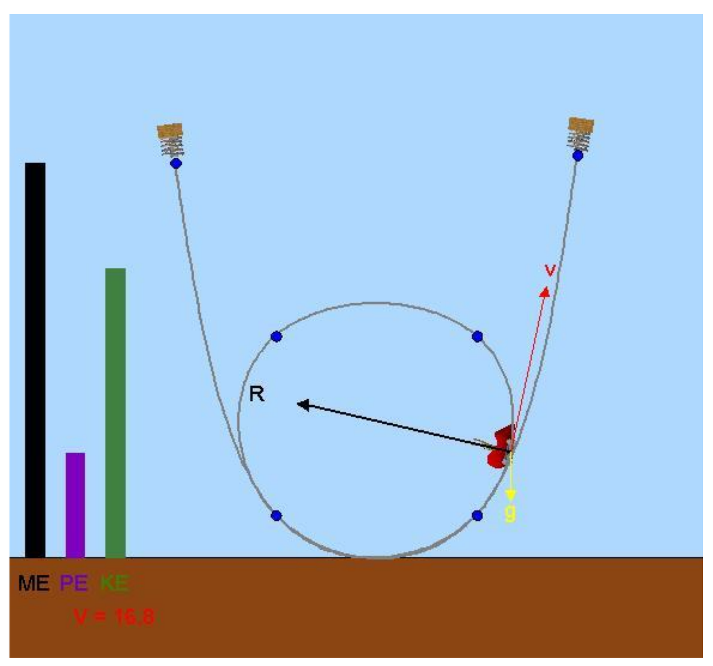
This poster presents the underlying physics used to simulate motions typically discussed under the auspices of "Roller Coaster Physics" in animation and an interactive java applet used in introductory physics. The car is modeled as a mas moving along a parametric curve, either at constant speed or coasting (with and without friction). Generating solutions to the resulting equations of motion places some constraints on the parametric equations describing the path of the track through space. The resulting materials are used to explore topics such as energy, power, circular motion, reaction forces and friction. In order to dramatize the repercussions of failing to loop at sufficient speed, collision dynamics are employed to simulate the car bounding off of the track and ground. Visual materials generated using the model are available online.(1,2)

Tracks

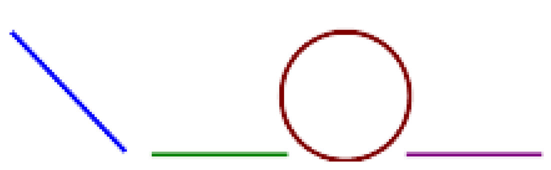
Storm Runner at Hershey Park, Pennsylvania

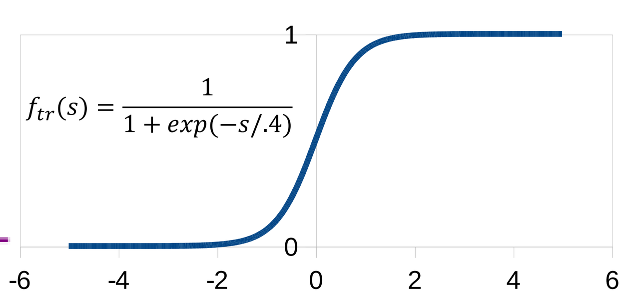
Parameterized paths in space: $\vec{r}(s)$

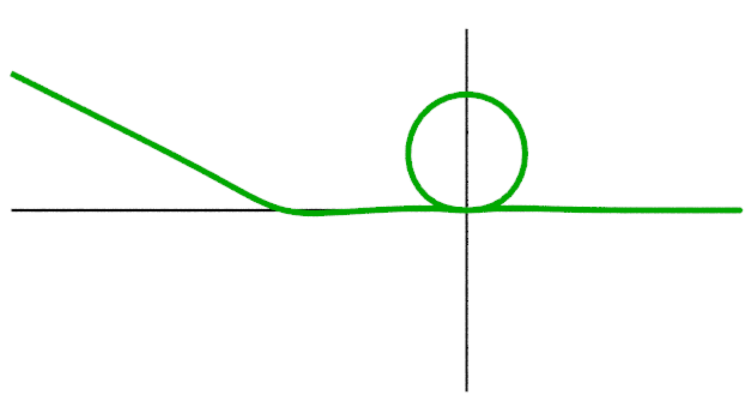
- Continuous
- 2^{nd} order differentiable



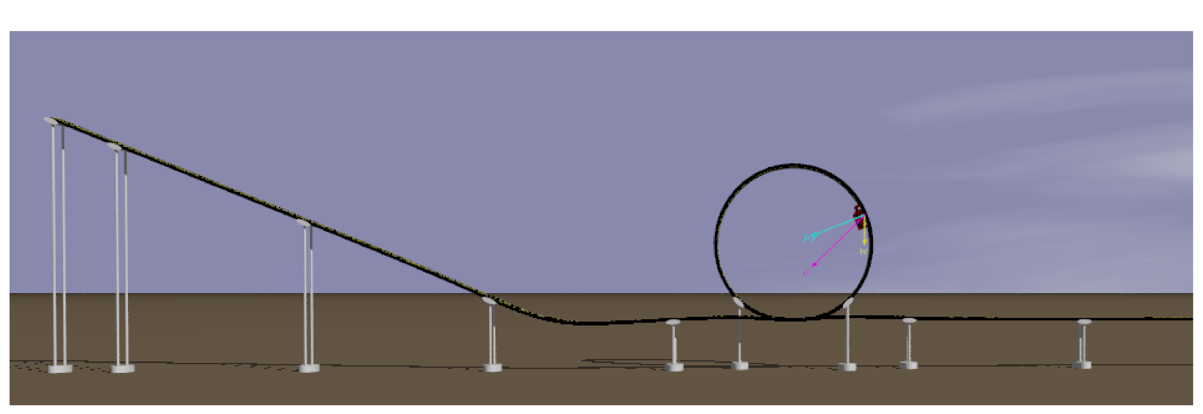
Applet: Cubic Spline Interpolation between adjustable control points.







Animation: functionally simple sections transitioned with a smooth "switch" function.



Dynamics 1: Constant Speed

(animation only)

Parameterized Path: $\vec{r}(s)$

Define: $\dot{\vec{r}} \equiv \frac{d\vec{r}}{dt}$ and $\vec{r}' \equiv \frac{d\vec{r}}{ds}$ then $\dot{\vec{r}} = \vec{r}'\dot{s}$

Speed $v = |\dot{\vec{r}}| = |\vec{r}'|\dot{s}$

Numerically integrate $ds = \frac{v}{|\vec{r}'|}dt$ to get $s(t), \vec{r}(s(t))$

Some useful derivative fun:

$$\dot{s} = \frac{v}{|\vec{r}'|} = v(\vec{r}' \cdot \vec{r}')^{-1/2}$$

$$\ddot{s} = \dot{v}(\vec{r}' \cdot \vec{r}')^{-1/2} - v^2(\vec{r}' \cdot \vec{r}')^{-2}\, \vec{r}' \cdot \vec{r}''$$

$$\ddot{s} = \dot{v}(\vec{r}' \cdot \vec{r}')^{-1/2} - v^2(\vec{r}' \cdot \vec{r}')^{-2}\, \vec{r}' \cdot \vec{r}''$$

$$\ddot{\vec{r}} = \vec{r}''\frac{v^2}{(\vec{r}' \cdot \vec{r}')} - \vec{r}'\left(\frac{\dot{v}}{(\vec{r}' \cdot \vec{r}')^{1/2}} - v^2\frac{\vec{r}' \cdot \vec{r}''}{(\vec{r}' \cdot \vec{r}')^2}\right)$$

Reaction Force: $\vec{F}_R + \mathrm{m}\vec{g} = m\ddot{\vec{r}}$

Perceived "g-force": $\frac{|\vec{F}_R|}{m\vec{g}}$

Dynamics 2.0: Coasting

Application of Lagrangian Dynamics(3): $L = T - V$

$$T = \frac{1}{2}mv^2 = \frac{1}{2}m(\vec{r}' \cdot \vec{r}')\dot{s}^2$$

$$V = mgy = mg\vec{r} \cdot \hat{\jmath}$$

From Lagrange's Equation $\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s} = 0$

$$m(\vec{r}' \cdot \vec{r}')\ddot{s} + m(\vec{r}' \cdot \vec{r}'')\dot{s}^2 + mgy' = 0$$

$$\ddot{s} = -\frac{(\vec{r}' \cdot \vec{r}'')\dot{s}^2 + gy'}{\vec{r}' \cdot \vec{r}'}$$

Integrate:

$$s(t + dt) = s(t) + \dot{s}(t)dt$$
$$\dot{s}(t + dt) = \dot{s}(t) + \ddot{s}(t)dt$$

Dynamics 2.1: Coasting with Friction

(applet only)

More Lagrangian Dynamics!

Rayleigh's dissipation function $\Phi = \frac{1}{2}bv^2$

Modified Lagrange's Equation $\frac{d}{dt}\frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s} + \frac{\partial \Phi}{\partial \dot{s}} = 0$

$$m(\vec{r}' \cdot \vec{r}')\ddot{s} + m(\vec{r}' \cdot \vec{r}'')\dot{s}^2 + mgy' + b(\vec{r}' \cdot \vec{r}')\dot{s} = 0$$

$$\ddot{s} = -\frac{(\vec{r}' \cdot \vec{r}'')\dot{s}^2 + gy'}{\vec{r}' \cdot \vec{r}'} - \frac{b}{m}\dot{s}$$

Fine-Tune Energy conservation:

$$dW_f = \vec{F} \cdot d\vec{r} = -b\vec{v} \cdot \vec{v}dt = -b(\vec{r}' \cdot \vec{r}')\dot{s}^2$$

$$E \to E + dW_f$$

Adjust speed each iteration to scale KE to "fix" Energy

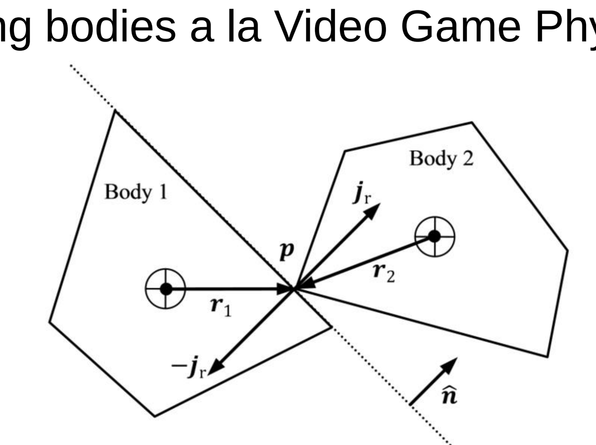
Dynamics 3: Bouncing

(animation only)

Consequences of "Failure to Loop"

Freefall between "bounces" from track

Collision Dynamics: Impulse approximation for interacting bodies a la Video Game Physics.(4)



- Conservation of momentum, angular momentum
- Coefficient of restitution e
- "Massive" body 1 $({m_1}^{-1} = 0,\ {I_1}^{-1} = 0)$ for track

Impulse:

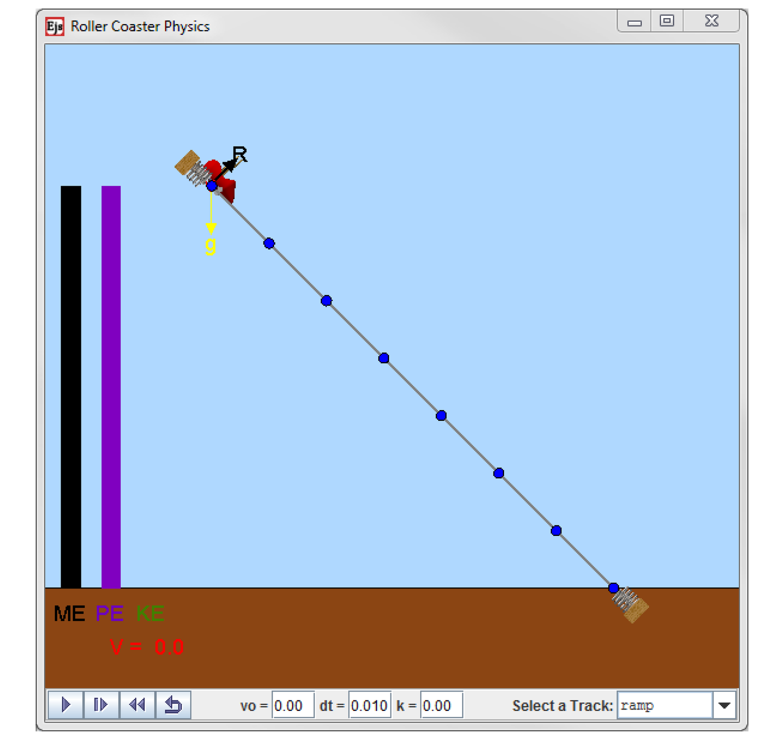
$$\vec{j}_r = -\frac{(1+e)\vec{v} \cdot \hat{n}}{{m_2}^{-1} + ({I_2}^{-1}(\vec{r}_2 \times \hat{n}) \times \vec{r}_2) \cdot \hat{n}}$$

$$\Delta\vec{v}_2 = \frac{\vec{j}_r}{m_2}\hat{n} \qquad \Delta\omega_2 = \vec{j}_r {I_2}^{-1}(\vec{r}_2 \times \hat{n})$$

Implemented in 2-D

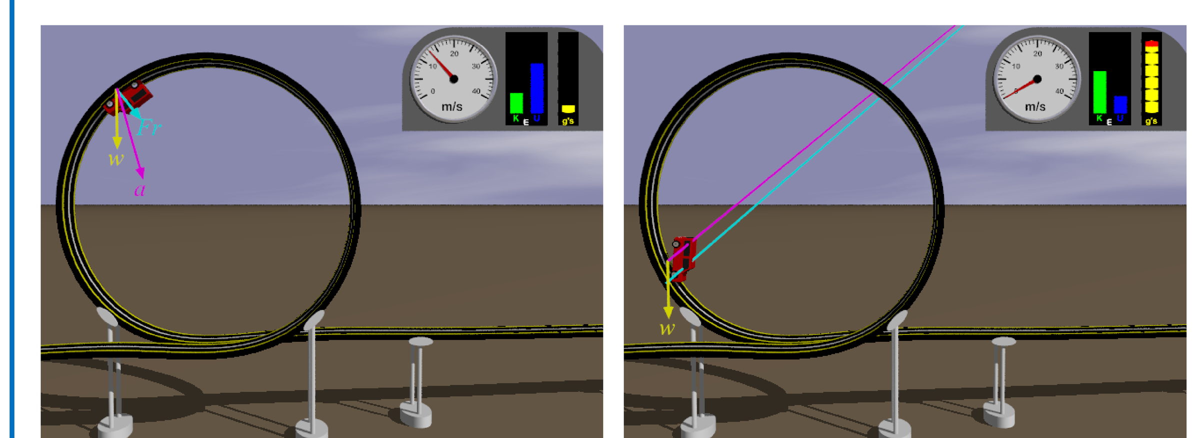
Implementation

Applet: Implemented Using Easy Java Simulations from the Open Source Physics Project(5)

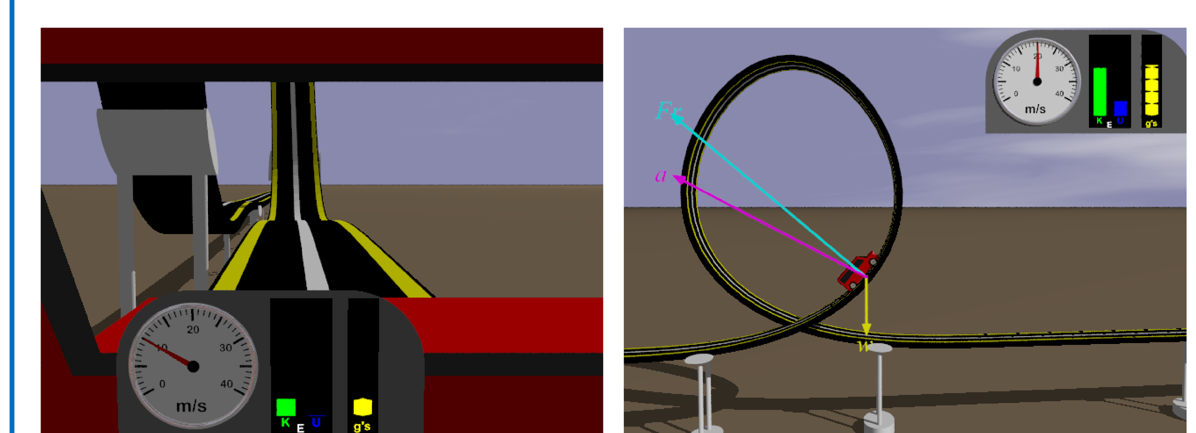


Interactive explorations of topics in energy conservation, reaction forces, friction and g-forces.

Animation: Frames created with POV-Ray 3-D ray tracing program.(6)



Coasting and Failing to Coast examples



First person views and non circular loop

With additional physical data (radius, mass, speed, initial height, etc) students tasked with determining:

- **Minimum speed to power through loop at constant speed**
- **Maximum power required during successful loop at constant speed**
- **G-forces, reaction forces**
- **Minimum height to coast through circular loop**

Some of this project is from *Animations for Physics and Astronomy* at Penn State Schuylkill. (7)

(1) Roller Coaster Model (Java applet) http://www.opensourcephysics.org/items/detail.cfm?ID=8228
(2) Roller Coaster Physics Animation http://www.youtube.com/watch?v=5yD2tOhI8SU
(3) See, e.g. Goldstein, Herbert (1980) *Classical Mechanics*.
(4) Vella, Colin. *Gravitas: An extensible physics engine framework using object-oriented and design pattern-driven software architecture principles* (Masters Thesis), Department of Communications and Computer Engineering, University of Malta, 2008.
(5) http://www.opensourcephysics.org
(6) http://www.povray.org
(7) http://phys23p.sl.psu.edu/phys_anim/Phys_anim.htm