Project 4: Numerical approximation of ODE’s – Roller Coaster Simulation

OVERVIEW
In this project, you will simulate the motion of a roller coaster along a parametrically defined track, in order to investigate Lagrangian mechanics and accuracy of different numerical methods for approximating solutions of ordinary differential equations (ODE’s).

BACKGROUND
In class, we examined using Lagrangian mechanics to derive the equations of motion for a dynamical system. We begin by defining $q$, the vector of $n$ generalized position coordinates for the system (typically one per degree of freedom of the system). The Lagrangian of a conservative (energy-preserving) system is defined to be

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q)$$

where $T(q, \dot{q})$ is the kinetic energy, and $V(q)$ is the potential energy. The Euler-Lagrange equation states that for each element $q_i$ of the generalized coordinates,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0$$

which gives $n$ second-order differential equations in the elements of $q$.

We then transform the resulting differential equations into a set of $2n$ equations by defining $y$ to be the state vector of the system consisting of both positions and velocities, with

$$y = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \quad \text{and} \quad \dot{y} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}$$

and subsequently numerically solving the ODE system

$$\dot{y} = f(y)$$

where the first $n$ elements of $f$ return the generalized velocities, and the second $n$ elements compute the generalized accelerations computed by the Euler-Lagrange equation.$^1$

$^1$Note that a general ODE allows a function $\frac{d}{dt}y = f(t, y)$. We omit the dependence of $f$ on $t$ to indicate that our system is time invariant – that is, the dynamics do not change over time.
TASKS

In class, we examined the case where the “roller coaster” is defined as a function $f(x)$, with the single generalized position coordinate $q = x$. Unfortunately, this parameterization of the problem is ill-equipped to handle vertical loops (a.k.a. loop-de-loops).

Define an alternative parameterization in terms of a single position coordinate $q = u$, with both the $x$- and $y$-coordinates of the roller coaster given by two parametric functions $x(u)$ and $y(u)$, and derive the corresponding equations of motion by forming the Euler-Lagrange equation for the new system.

Design a roller coaster by defining the two functions $x(u)$ and $y(u)$. Your roller coaster must include at least two hills and one vertical loop. If you’re looking for information about designing parametric curves, it might be helpful to examine the Wikipedia page on Hermite interpolation: http://en.wikipedia.org/wiki/Hermite_interpolation.

Assume that both $x$ and $y$ are expressed in meters, and design your roller coaster to have a plausible physical scale. You should set the initial conditions such that the roller coaster has enough energy to get from $u = 0$ to the end of the roller coaster.

Simulate your roller coaster by using both Euler’s method and one second order (or better) Runge-Kutta method, such as Heun’s method or the midpoint method. Simulate each algorithm using a timestep of 0.01s and 0.001s, for a total of four simulation runs. For each simulation, produce an animation of the result. To prevent excessively large animations, only produce a frame for every fourth step when using 0.01s timesteps, and every 40 steps for 0.001s. For information on how to produce GIF animations with MATLAB, see http://www.mathworks.com/matlabcentral/fileexchange/21944-animated-gif/content/AnimatedGIF/html/AnimatedGif.html.
WHAT TO TURN IN

You should submit all of your source code, animations, and a PDF writeup addressing the following:

• Show that your derivation of the Euler-Lagrange equation is a generalization of the case we derived in class by setting \( y(u) = f(u) \) and \( x(u) = u \).

• For each numerical integration scheme and step size, produce a plot graphing the kinetic energy, potential energy, and total energy (i.e. their sum) over time. You should have 4 plots with three traces each.

• Noting that the total energy should be constant over time (since the system is conservative), interpret the plots in terms of the effectiveness of each integration scheme and stepsize.

• Include remarks on any unexpected or surprising behavior of your simulations, as well as any pitfalls you encountered in implementing this project.

Submit all materials via email.