

Newton's Cradle

by Glória de Sá Pereira, 81826 & João Cunha, 81541

January 2015

1 Introduction

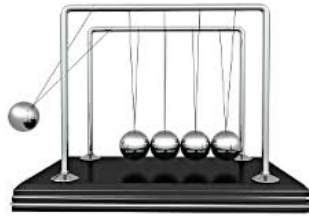


Figure 1: Newton's Cradle.

Newton's cradle, named after Sir Isaac Newton, is a device that demonstrates conservation of momentum and energy via a series of collisions between swinging spheres. When the one, or more, is lifted and released, it strikes the stationary spheres; a force is transmitted through those spheres, pushing the last one upward.

With this program, we hope to reproduce computationally the movement of an ideal cradle. This means we will consider a system where both the energy and momentum of the cradle are conserved, which makes it easier to understand its motion. In order to describe the cradle's motion we will use the Euler-Cromer method of solving equations, which is more precise than the usual numeric method, and better adapted for programming.

2 Theoretical Explanation

If a sphere is pulled away from the others, it stores a certain amount of potencial energy, proporcional to its heigth. When released, it falls due to the effect of gravity, transforming its potencial energy into kinetic energy. Eventually, it collides with the stationary ones, immediately ceasing its movement and transfering all its velocity to the sphere with whom it collides (and therefore all its momentum and energy). This will happen successively, sphere after sphere, until the moment it reaches the last one, which will then begin its upward movement with the same velocity that the first sphere had when it collided with the stationary ones, thus reaching the same height from which that first one was released. If more than one sphere is released, all happens in the exact same way, so that the laws of phisycs are respected.

3 Program Functioning

After sucessfully compilling the program with `./ggtk`, an initial window will appear. That window will be the core of the program, allowing the user to define a series os characteristics about the cradle, with buttons that define:

- The number of spheres initially released from both sides;
- Initial velocity of the spheres;
- Initial angle of the spheres.

While creating this project, our main goal was to create a simple and interative program that could easily be used by anyone, much as maintaining its scientific correction. The first thing the user will have to do when using the program, will be to decide how many spheres he wants the cradle to have. This can be done by two different means; he can either type the number of pendulums he wants, directly into the respective box, or he can keep adding them, by clicking on the arrows, up or down.

Total Number of Pendulums	<input type="text" value="3.00"/>	<div>↑ ↓</div>
---------------------------	-----------------------------------	--------------------

Note that, to change the other variables, the procedure will be the same. The default number of pendulums appearing in the initial

window will be of three. The maximum number of pendulums will be fifteen, and the minimum two. Then, the user will have to decide the number of spheres being released from each side of the cradle, knowing that the sum of spheres leaving the left and the right side, cannot be bigger than the total number of spheres in the pendulum.

Left	
Number of pendulums that are released	0.00 <input type="button" value="+"/> <input type="button" value="-"/>
Right	
Number of pendulums that are released	0.00 <input type="button" value="+"/> <input type="button" value="-"/>

If, by mistake, the user doesn't respect this, a new window will open, alerting him of his error.
At this point, the only thing left to do before being able to run the program is giving values to the initial velocity and/or angle of the spheres on both sides of the cradle.

Left	
Velocity (m/s)	0.00 <input type="button" value="+"/> <input type="button" value="-"/>
Angle (°)	0.00 <input type="button" value="+"/> <input type="button" value="-"/>

After this, the user will finally be ready to see the motion of the cradle he "built". To do so, he will only have to press the *play* button, and the pendulum will start its movement. A *pause* button is also available, so that the user can freeze the movement at any moment. Finally, we created a *reset* button, that, when pressed, brings every variable to the initial values. To quit the program, the user can press the *quit* button, or the usual quitting button on the left right corner. There is also a *about* button that gives information about this program and its writers. To give some extra customization to the cradle, we included some background themes that can be selected by the user in the top right of the window.

Reset	Pause	Play	Quit
Themes:			
Linux	The Beatles	Default	Miley