A Graphical User Interface to Explore the Brachistochrone Problem

Jessica Flores Rosado
Department of Mathematics
University of Puerto Rico, Humacao

Mayo 25, 2006

I   Introduction

The Brachistochrone Problem was posed by Johann Bernoulli in June 1696, as a challenging problem to the most brilliant mathematicians in the world. The problem was the following:

> *Given two points* A *and* B *in a vertical plane, what is the curve traced out by a point acted on only by gravity, which starts at* A *and reaches* B *in the shortest time.*

Among the mathematicians that solved this problem were: Isaac Newton, Jacob Bernoulli, Leibniz, de L'Hôpital, and of course Johann Bernoulli.

Common sense would tell us that the fastest way to travel from a point A to a point B in a vertical plane is through a straight line from point A to B. But the actual solution to this problem is a curve called a *cycloid*.

In this project we developed a program (a graphical user interface) in which the user can experiment with different types of curves and determine the corresponding times of descend along them. The user can choose from various families of curves and see an actual animation of a bead through that curve and see how long it took to descend it.

II   Work Preformed

We used MATLAB to implement a graphical user interface (GUI) to experiment with the Brachistochrone Problem. The GUI includes a menu from which the user can choose from four different kinds of curves: straight line, parabola, exponential, and a cycloid. Each curve has different attributes or parameters, with which the user can experiment. The interface has a button that runs a bead through the curve that the user chose. After the bead runs through that curve, the time that it took to travel from the point (0,0) to point (a,b) appears on the interface. Then the user can keep choosing other curves to see which curve has the shortest time.

III   Program User Interface

The interface has a menu with four different kinds of curves: the parabola, the cycloid, the exponential, and the straight line. The user can choose between these four curves, and then input certain attributes of that curve. A message will appear for the user so that the user knows which attributes are needed for that curve. These attributes are the point in which the curve is going to end, the initial velocity of the bead, between others. Then the user can press a button that will run the bead through the chosen curve. After this, the user will see the results of the time that it took for the bead to get to the point that the user had input on the attributes. Then the user can choose between the other curves, using the same velocity and end point to see which curve is the fastest. As the user will soon discover, the cycloid will always have the shortest run time. In Figure A we show a sample of the graphical user interface.
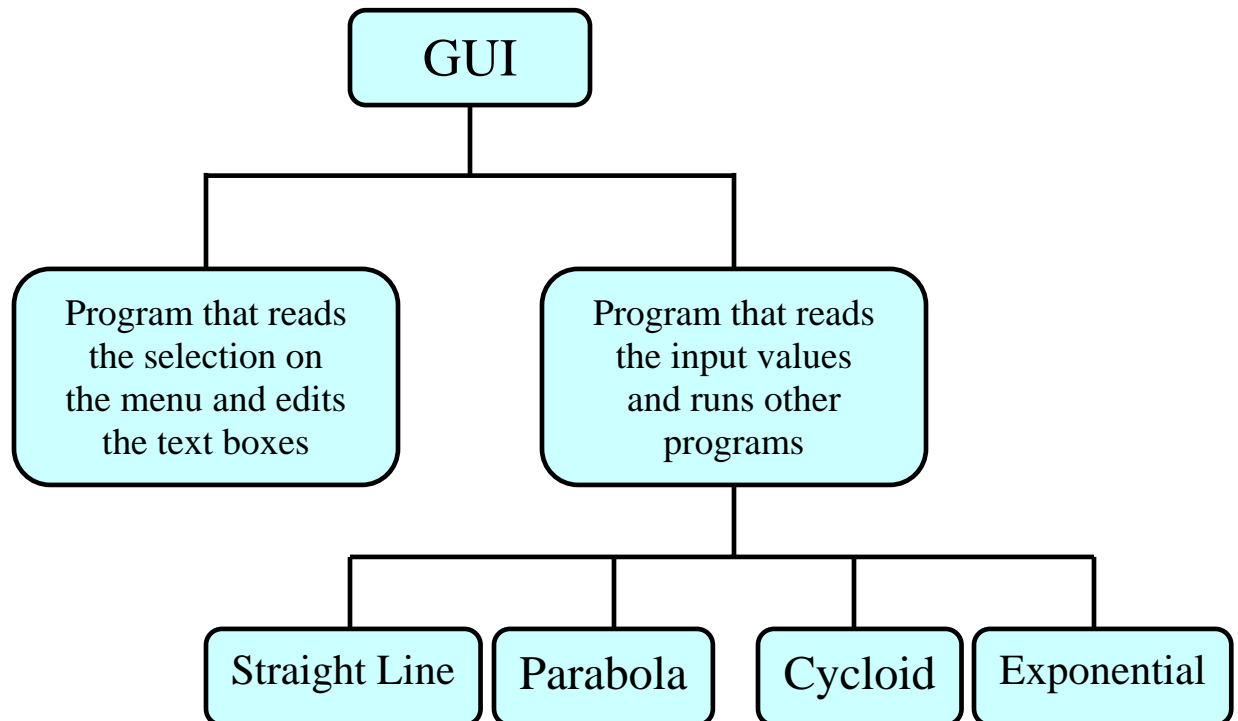
Figure A:
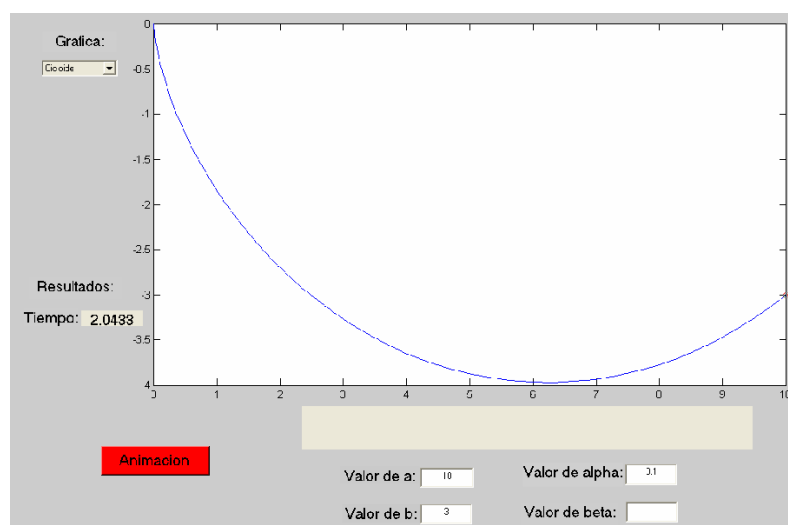
IV   Internal Composition of GUI

The internal composition of GUI consists of three principal modules. The first one is the user interface, which has a menu with four different kinds of curves and the user can select one of them. The user then receives a message through a text box, which indicates the attributes the user must input for the curve chosen. This attributes are inputted in an editable text box, which will be read by another program that will determine if these attributes are correct. If not, the user will receive an error message through a text box. After the user enters the correct information, the user can press a button that will run one out of four programs, depending on the curve selected on the menu, that will graph the curve and perform the bead animation. Each of the four programs calculate the time that it takes for the bead to get to the point (a,b), so the user will be able to see how long it took, through a text box. In Figure B we show a diagram of the flow in the internal composition of GUI.

Figure B: