Using the Bunimovich Stadium Evolution Viewer Software

1. To use the Bunimovich Stadium Evolution Viewer you must first run the *BunimovichStadiumViewer-GUI.py* file. If you are using windows 10 you can do this by clicking on the file named *BSE GUI.exe*. Once the software is running the window in Figure 1 should pop up.

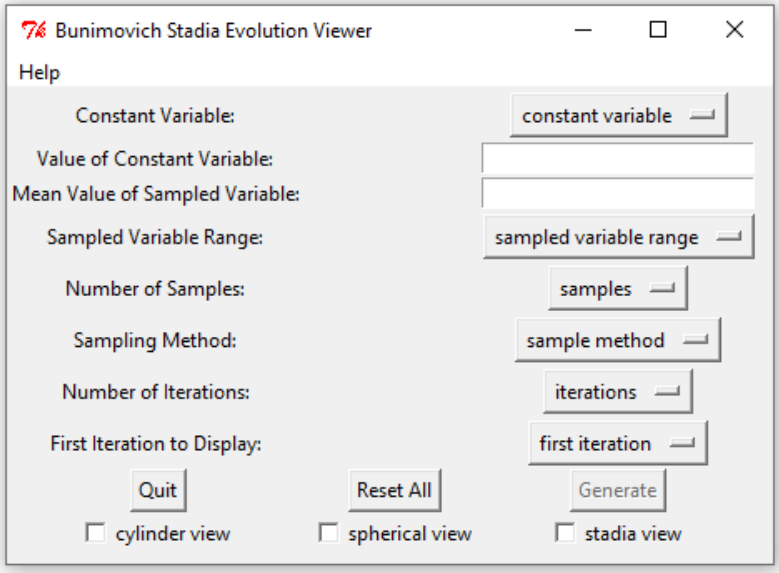

Figure 1: The graphical user interface (GUI) for the Bunimovich Stadium Evolution Viewer Software

2. To use the software fill in all of the fields and click the *Generate* button to produce pdf(s). Please note that all fields must have a value and atleast one of the check boxes must be checked before *Generate* is clickable. For an explanation of all of the fields and the software please see below.

Explanation of the Software

This software allows the user to see how the orbits of nearby billiards in the Bunimovich stadium evolve over time. A set of nearby points in the collision space is the input of the program. The output is visual data representing the successive collisions of the points in the input set. Recall that in the Bunimovich stadia the collision space is represented by points of the form (θ, φ). θ is the polar angle of the collision point on the stadium's boundary. φ is the angle between the inward normal $\hat{N}$ of the stadium at θ and the post-collisional velocity vector. The two parameters have the following ranges $\theta \in [0, 2\pi)$ and $\varphi \in (-\frac{\pi}{2}, \frac{\pi}{2})$. Graphical representations of θ and φ can be seen in Figure 2.

The input sets can be of two different forms:

(1) A set of collision points with θ constant and φ varying, or

(2) A set of collision points with φ constant and θ varying.

Once the inputs are specified the software can produce 3 different kinds of views of the system's evolution over successive applications of Bunimovich stadium's collision map. The first view is the trajectory of the

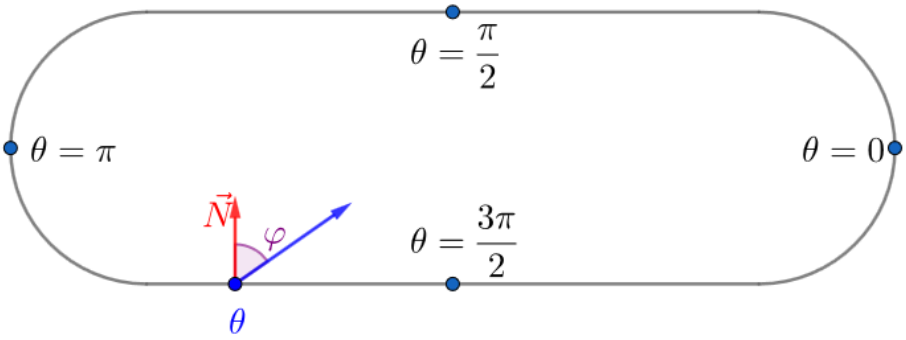

Figure 2: The inward normal $\hat{N}$, φ, and θ

billiards corresponding to the input set "bouncing" in the stadia. To understand the second view notice that the collision space is homeomorphic to a cylinder since θ is cyclic and φ is in an interval. Thus, the second view shows the billiards' trajectories represented on the cylinder[1]. The third view is the collision space projected on a sphere. Once the inputs have been supplied and the *Generate* button is clicked, pdf(s) of the desired views are produced in the directory containing the .py or .exe file.

Providing Input to the Software

There are eight fields of input that must be supplied to the software in addition to the check boxes that specify which views to produce. For an explanation of a particular input field please find its label below.

Constant Variable This specifies which variable is to be constant, either θ (theta) or φ (phi).

Value of Constant Variable This sets the value of the constant variable. To supply the proper input recall that $\theta \in [0, 2\pi)$ and $\varphi \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

Mean Value of Sampled Variable This, along with *Sampled variable Range* specifies the range of values that the non-constant variable is allowed to take. For instance if 1 is specified as the *Mean Value of Sampled Variable* and "pi/8", (about 0.392) is specified as *Sampled variable Range* then the range of the non-constant variable would be (about) 1 ± 0.392.

Sampled Variable Range (Please see the above explanation of *Mean Value of Sampled Variable*).

Number of Samples The software can not compute the trajectory of every point in the ranges specified by the user. Instead it takes a sample of points in the range. This field tells the software how many points to sample. The more points that are sampled, the more accurate the output is.

Sample Method This specifies the sampling technique to be used by the software. If "even" is selected then the points are evenly spaced in the non-constant variable's range. If "random" is selected then points in the specified range are chosen according the the uniform distribution.

Number of Iterations This specifies the number of iterations of the collision map to be applied to the sampled points.

First iteration to Display This specifies the first iteration to be displayed in the output pdf. For example if *Number of Iterations* is set to "10" and *First iteration to Display* is set to "5" then iterations 5-10 of the collision map will be displayed in the pdf(s). Please note that this value must be less than the value of *Number of Iterations*.

Checkboxes The check boxes specify which views will be displayed in the pdf(s). At least one of these must be checked, however the user can check as many as is desired. A pdf is created for each box checked in the same directory as the .exe or .py file.

[1]The cylinder is "cut" and "flattened out" so it can be easily viewed on a flat screen.