

Billar de Bunimovich

Iñigo Agirre Garrido

December 2022

1 Introduction

Leonid Abramowitsch Bunimovich (nacido el 1 de agosto de 1947) es un matemático soviético y estadounidense, que realizó aportes fundamentales a la teoría de los Sistemas Dinámicos, la Física Estadística y diversas aplicaciones. Bunimovich recibió su licenciatura en 1967, maestría en 1969 y doctorado en 1973 de la Universidad de Moscú.

Bunimovich es conocido principalmente por el descubrimiento de un mecanismo fundamental del caos en los sistemas dinámicos llamado mecanismo de desenfoque. Este descubrimiento fue una gran sorpresa no solo para la comunidad matemática sino también para la física. Los físicos no podían creer que tal fenómeno (¡físico!) fuera posible (a pesar de que se proporcionó una prueba matemática rigurosa) hasta que realizaron experimentos numéricos masivos. La clase más famosa de sistemas dinámicos caóticos de este tipo, los billares dinámicos, son billares caóticos enfocados como el estadio Bunimovich.

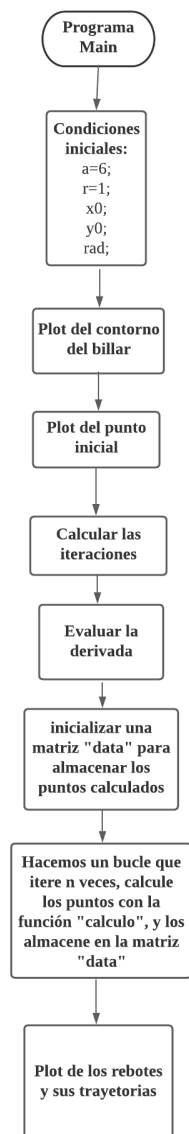
Se va a estudiar el caso particular de el estadio o billar de Bunimovich.

2 Código de el Billar de Bunimovich

El código realizado para modelizar el caos de un billar de Bunimovich se encuentra en mi página de github, en el siguiente repositorio:

Link al repositorio: <https://github.com/Agirre8/Billar-de-Bunimovich>

2.1 Diagrama de flujo del programa

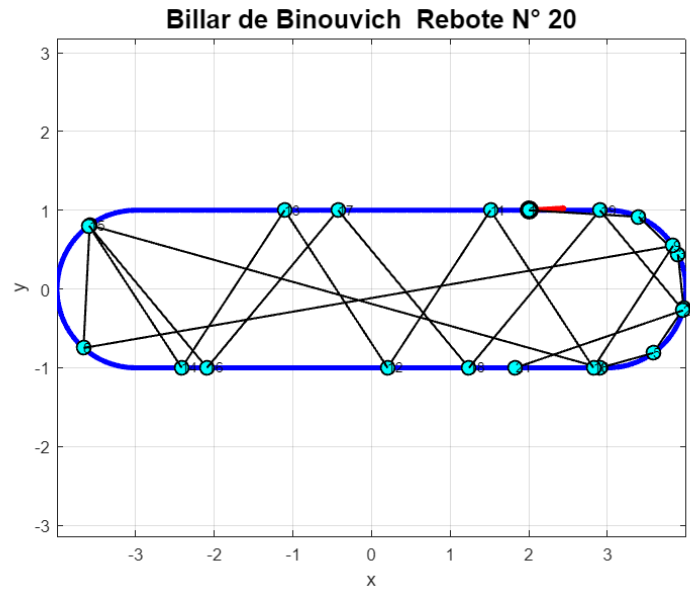


3 Ejecución del programa

Dadas tres condiciones iniciales, comprobar el funcionamiento del programa para cada una de estas:

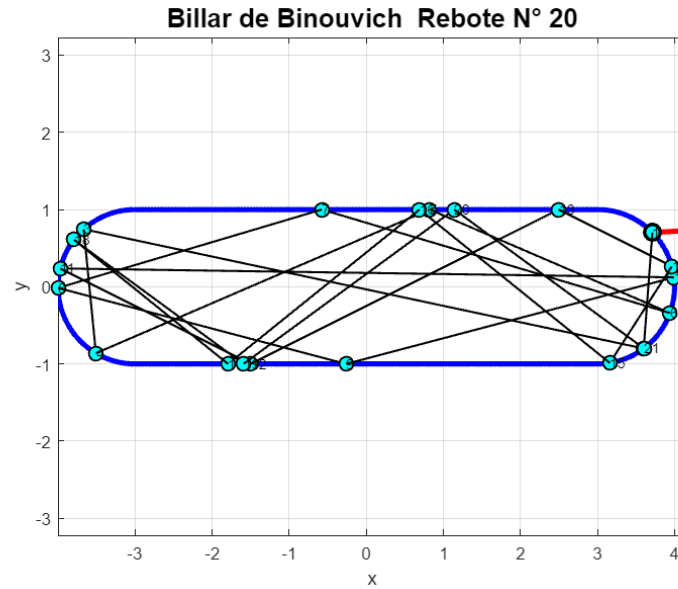
3.1 Primera condición inicial

$S_1 = (2, r, 3.4)$ siendo $S = (x, y, r)$ calcula y representa las trayectorias para 20 rebotes siendo $a=6$ y $r=1$.



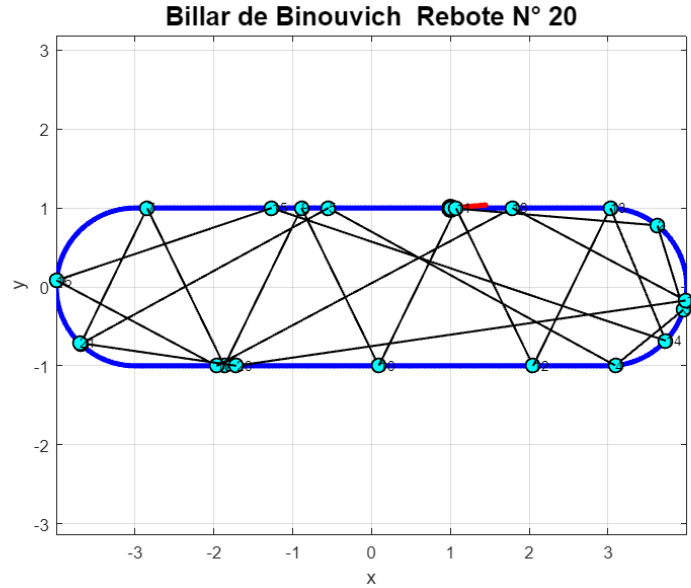
3.2 Segunda condición inicial

$S_2 = (3.7, r, 3.4)$ siendo $S = (x, y, r)$ calcula y representa las trayectorias para 20 rebotes siendo $a=6$ y $r=1$.



3.3 Tercera condición inicial

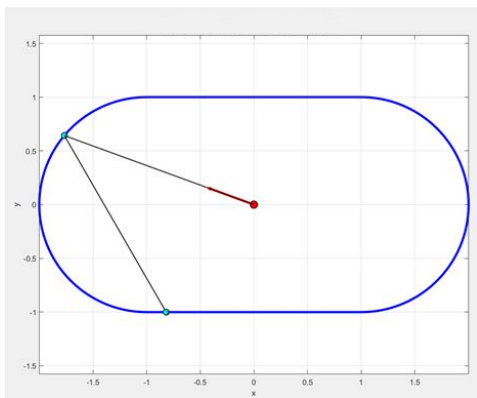
$S_3 = (1, r, 4.8)$ siendo $S = (x, y, r)$ calcula y representa las trayectorias para 20 rebotes siendo $a=6$ y $r=1$.



4 Cuestiones

4.1 Como se calcula la intersección en la curva

El programa busca posibles puntos de intersección con la curva y después calcula la tangente a ese punto para buscar el rebote, como el valor normal a la tangente del punto de intersección.



```

17 % Cuando punto esta sobre la linea
18 if table(j,5)==1
19     root=fzero(fun, (a+b)/2); % Usa el valor promedio de a y b para iniciar
20     if root>a & root<b % Verifica si existe una raiz o solución
21         z=[z,root]; % si lo hay agrega una posible solución a z
22     end
23
24 % Cuando el punto esta sobre la zona curva o elíptica
25 % xx1,xx2,yy1,yy2 son coordenadas de la intersección de la línea con el arco del círculo
26 elseif table(j,5)==2
27     % busca las coordenadas X Y del centro del radio del círculo
28     [cx,cy,rx,ry]=ellipseparam(table(j,1),table(j,2),table(j,3),table(j,4));
29     hit=0; %indica si la trayectoria esta sobre el arco o no
30     if abs(mod(ao,pi)-pi/2)<10^-10
31         if cx-rx<xo & xo<cx+rx
32             hit=1;
33             xx1=xo;
34             xx2=xo;
35             yy1=-sqrt(rx^2*(1-xx1^2/ry^2));
36             yy2=sqrt(rx^2*(1-xx2^2/ry^2));
37         end
38     else
39         m=tan(ao); %encuentra la pendiente de la trayectoria
40         b0=-m*xo+m*cx+yo-cy; % y-intercepción de la trayectoria y-(yo-cy)=m(x-(xo-cx))
41         aq=rx^2*m^2+ry^2; % a en la formula cuadratica para la raiz o solución
42         bq=2*m*b0*rx^2; % b en la formula cuadratica para la raiz o solución
43         cq=b0^2*rx^2-rx^2*ry^2; % c en la formula cuadratica para la raiz o solución
44         if bq^2-4*aq*cq>=0 % Verifica el determinante para ver si encuentra

```

4.2 Distinción de donde va a chocar la pelota

El programa, por cada rebote evalúa hacia donde se dirige la esfera e identifica entre dos opciones, que este sobre la línea recta o sobre el semicírculo. En función de eso decide que cálculos hace. "table(j,5)" almacena dos valores, 1 si está en la parte lineal y 2 si está sobre la parte curva.

Para saber que valores almacenar, si 1 o 2, se realizará la comparación entre dos puntos cercanos a X e Y para saber si hay puntos iguales. Si hay puntos iguales en Y indicaría que estamos en la recta mientras que si en ambos casos hay puntos distintos indicará que estamos en la curva.

4.3 Cálculo de los puntos de intersección

```
else
    m=tan(ao); %encuentra la pendiente de la trayectoria
    b0=-m*xo+m*cx+yo-cy; % y-intercepción de la trayectoria y-(yo-cy)=m(x-(xo-cx))
    aq=rx^2*m^2+ry^2; % a en la formula cuadratica para la raiz o solución
    bq=2*m*b0*rx^2; % b en la formula cuadratica para la raiz o solución
    cq=b0^2*rx^2-rx^2*ry^2;% c en la formula cuadratica para la raiz o solución
    if bq^2-4*aq*cq>=0 % Verifica el determinante para ver si encuentra
        hit=1; % trajectory hits arc
        xx1=(-bq-sqrt(bq^2-4*aq*cq))/2/aq; %x-coordenada de intersección 1
        xx2=(-bq+sqrt(bq^2-4*aq*cq))/2/aq; %x-coordenada de intersección 2
        yy1=m*xx1+b0; %y-coordenada de intersección 1
        yy2=m*xx2+b0; %y-coordenada de intersección 2
    end
```