

Análisis del mapa de Poincaré en el Oscilador de Duffing

Juan José Ochoa Duque, Santiago Andrés Pérez Acevedo, Bryan Pérez Múnera.

Instituto de Física

Universidad de Antioquia

5 de junio de 2020

Resumen

En el siguiente trabajo se muestra un estudio del caos que presenta el sistema de Duffing a partir de una de sus secciones de Poincaré. Se parte inicialmente del sistema de Duffing, asignándole valores determinados a sus parámetros. Implementando una integración usando RK4 se obtienen las soluciones del sistema para un determinado tiempo y finalmente a partir de dichas soluciones se halla una de sus secciones de Poincaré. Lo anterior siendo implementando en C++, con el uso de Python para la parte gráfica.

I. INTRODUCCIÓN

Los sistemas físicos ideales son ficciones sobre las cuales se modelan aquellos sistemas físicos que describen la naturaleza. Como sistemas ideales, tienen la característica de que tienen soluciones en general analíticas y estables, es decir que permiten un gran margen de predictibilidad a la hora de ser estudiados. La naturaleza en sí misma, ya en un marco más general, puede describirse a partir de los sistemas dinámicos. Dentro de estos surge el concepto de caos, que en resumidas palabras indica cuándo un sistema deja de ser estable en sus soluciones y pierde todo margen de predictibilidad. Para determinar el caos en un sistema dinámico hay varios criterios tanto cualitativos como cuantitativos, unos más exactos y precisos que otros, que se pueden usar. Los mapas o secciones de Poincaré es uno de dichos criterios, de carácter cualitativo, pero de gran fiabilidad para determinar el caos en un sistema.

II. MARCO TEÓRICO

i. Sistema Dinámico

Si el estado de un sistema es definido por las n variables $X_i = X_i(t)$, este se puede describir como $\vec{X} = \{X_1 \dots X_n\}$ donde se define su evolución temporal como $\dot{\vec{X}} = \frac{d\vec{X}}{dt} = \vec{f}(\vec{X}, t, \alpha)$, que en términos de un sistema de ecuaciones diferenciales toma la siguiente forma:

$$\begin{cases} \dot{X}_1 = \frac{dX_1}{dt} = f_1(\vec{X}, t, \alpha) \\ \vdots \\ \dot{X}_n = \frac{dX_n}{dt} = f_n(\vec{X}, t, \alpha) \end{cases}$$

Lo que se conoce como sistema dinámico. Las anteriores ecuaciones me dan cuenta entonces de cómo evolucionan las n variables X_i de mi sistema en el tiempo. Donde los parámetros α son característicos de cada sistema y la evolución del sistema se dan a partir de las condiciones iniciales.

ii. Retrato de Fase, Mapa de Poincaré y Caos

En el estudio de un sistema dinámico se busca qué trayectorias u órbitas específicas sigue un

determinado sistema. Al conjunto de órbitas o trayectorias de la evolución del sistema, en el espacio abstracto de sus coordenadas generalizadas y sus respectivos momentos canónicos conjugados (espacio de fase), se le conoce como su retrato de fase. Donde cada estado específico del sistema en un tiempo t es un punto en este y su evolución determina entonces el retrato de fase.

El mapa de Poincaré es un plano que corta el retrato de fase en una posición arbitraria. Este mapa da cuenta de cuántas veces en un determinado tiempo el retrato de fase lo corta, dejando registro entonces de qué tanto varían las trayectorias del retrato de fase en el tiempo. Un sistema dinámico presenta caos cuando sus soluciones dejan de ser estables, es decir, cuando no se tiene predictibilidad del comportamiento en el espacio de fase de las trayectorias del sistema a través del tiempo. Así mismo, cuando presenta alta sensibilidad a las condiciones iniciales, en el sentido de que cuando se hacen pequeñas variaciones de las condiciones iniciales se producen grandes diferencias en el retrato de fase, se dice que hay posible caos en el sistema. Los mapas de Poincaré al determinar qué tanto difieren las trayectorias a través del tiempo, es otro buen criterio e indicativo de un comportamiento caótico del sistema. Entre más se llene de intersecciones el mapa de Poincaré con las trayectorias, más caótico es el sistema.

iii. Ecuación de Duffing

La ecuación de Duffing viene dada por la siguiente ecuación diferencial de segundo orden, no lineal e inhomogénea:

$$\ddot{x}(t) + \delta\dot{x}(t) + \beta x(t) + \alpha x^3(t) = \gamma \cos(\omega t)$$

Esta representa en general diversos osciladores forzados, amortiguados con un forzamiento periódico, donde los parámetros constantes δ controla la cantidad de amortiguamiento del sistema, β controla la rigidez lineal del sistema, α es el parámetro de la no linealidad, γ es la amplitud del forzamiento, ω es la frecuencia angular de dicho forzamiento y la variable

$x(t)$ el desplazamiento en función del tiempo. Notando que si se toma la primera derivada temporal del desplazamiento como una variable del sistema junto a al desplazamiento y la derivada temporal de la frecuencia ω , se puede obtener la forma de un sistema dinámico de la forma siguiente:

$$\begin{cases} \frac{d}{dt}x = \dot{x} \\ \frac{d}{dt}\dot{x} = -\delta\dot{x} - \beta x - \alpha x^3 + \gamma \cos(\omega t) \\ \frac{d}{dt}\phi = \omega \end{cases}$$

Este es el sistema dinámico de Duffing. Para los valores de $\beta > 0$ el sistema da cuenta de un oscilador forzado con una fuerza restauradora $F = -\beta x - \alpha x^3$ y para el caso de $\beta < 0$ el sistema da cuenta de la dinámica de una masa puntual en un doble pozo de potencial. Es para este caso en el que puede haber caos en el sistema.

Como ilustración de la dinámica del sistema en el caso particular $\gamma = 0$ y $\alpha > 0$ se presenta la figura 1, que da cuenta de la forma del potencial y las trayectorias sobre este según los valores de δ y β .

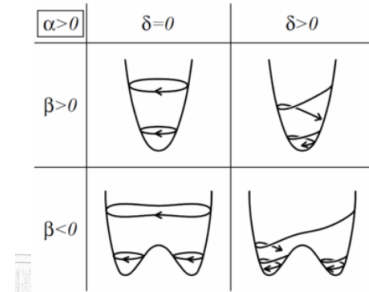


Figura. 1: Pozos de potencial para un forzamiento nulo y $\alpha > 0$

Donde notamos las variaciones de la trayectoria debido al parámetro de amortiguación δ y las variaciones del potencial debido al parámetro β .

En el caso en el que se consideran todos los parámetros del sistema (una masa puntual en un pozo doble de potencial), hay una semejanza en la dinámica de una tira de metal sometida a la fuerza magnética producida por dos imanes, así mismo sometida a una

fuerza externa sinnusoidal. En la figura 2 se muestra la situación descrita.

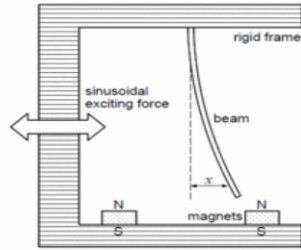


Figura. 2: Tira metálica sometida a dos imanes y fuerza sinusoidal externa

III. IMPLEMENTACIÓN Y RESULTADOS

Se implementaron dos códigos en C++, uno encargado de la integración a partir del método RK4 y otro encargado de la construcción del mapa de Poincaré. Esto para los valores de los parámetros $\alpha=1$, $\beta=-1$, $\delta=0.2$, $\gamma=0.1$, $\omega=1$. En la figura 3 se muestra el código que construye el mapa de Poincaré. Este recibe los datos del retrato de fase, es decir, de las soluciones del integrador, que se muestra en la figura 4.

```

1 #include<iostream>
2 #include<iomanip>
3 #include<cmath>
4 #include<fstream>
5 #include<cstdlib>
6
7 using namespace std;
8 void leerArchivo(string& k)
9
10 int main()
11     string nombre_archivo;
12     nombre_archivo="datos.dat";
13     leerArchivo(nombre_archivo);
14
15 void leerArchivo(string& archivo){
16     ifstream p;
17     p.open(archivo.c_str());
18     if(p.is_open())
19         //Comprobamos si el archivo existe
20         if(p.is_open())
21             cout<<"El archivo no se abrió" con éxito<<endl;
22             exit(1);
23         //Formato de los datos
24         double matriz[10000][3];
25         for(int i=0;i<10000;i++){
26             for(int j=0;j<3;j++){
27                 matriz[i][j]=p.get();
28             }
29         }
30         p.close();
31     }
32     //Formato de los datos
33     double l1[N];
34     double l2[N];
35     for(int i=0;i<N;i++){
36         l1[i]=matriz[i][0];
37         l2[i]=matriz[i][1];
38     }
39     //Formato de los datos
40     double l3[N];
41     double l4[N];
42     for(int i=0;i<N;i++){
43         l3[i]=matriz[i][2];
44         l4[i]=matriz[i][3];
45     }
46     //Formato de los datos
47     double l5[N];
48     double l6[N];
49     for(int i=0;i<N;i++){
50         l5[i]=matriz[i][4];
51         l6[i]=matriz[i][5];
52     }
53     //Formato de los datos
54     double l7[N];
55     double l8[N];
56     for(int i=0;i<N;i++){
57         l7[i]=matriz[i][6];
58         l8[i]=matriz[i][7];
59     }
60     //Formato de los datos
61     double l9[N];
62     double l10[N];
63     for(int i=0;i<N;i++){
64         l9[i]=matriz[i][8];
65         l10[i]=matriz[i][9];
66     }
67     //Formato de los datos
68     double l11[N];
69     double l12[N];
70     for(int i=0;i<N;i++){
71         l11[i]=matriz[i][10];
72         l12[i]=matriz[i][11];
73     }
74     //Formato de los datos
75     double l13[N];
76     double l14[N];
77     for(int i=0;i<N;i++){
78         l13[i]=matriz[i][12];
79         l14[i]=matriz[i][13];
80     }
81     //Formato de los datos
82     double l15[N];
83     double l16[N];
84     for(int i=0;i<N;i++){
85         l15[i]=matriz[i][14];
86         l16[i]=matriz[i][15];
87     }
88     //Formato de los datos
89     double l17[N];
90     double l18[N];
91     for(int i=0;i<N;i++){
92         l17[i]=matriz[i][16];
93         l18[i]=matriz[i][17];
94     }
95     //Formato de los datos
96     double l19[N];
97     double l20[N];
98     for(int i=0;i<N;i++){
99         l19[i]=matriz[i][18];
100         l20[i]=matriz[i][19];
101     }
102     //Formato de los datos
103     double l21[N];
104     double l22[N];
105     for(int i=0;i<N;i++){
106         l21[i]=matriz[i][20];
107         l22[i]=matriz[i][21];
108     }
109     //Formato de los datos
110     double l23[N];
111     double l24[N];
112     for(int i=0;i<N;i++){
113         l23[i]=matriz[i][22];
114         l24[i]=matriz[i][23];
115     }
116     //Formato de los datos
117     double l25[N];
118     double l26[N];
119     for(int i=0;i<N;i++){
120         l25[i]=matriz[i][24];
121         l26[i]=matriz[i][25];
122     }
123     //Formato de los datos
124     double l27[N];
125     double l28[N];
126     for(int i=0;i<N;i++){
127         l27[i]=matriz[i][26];
128         l28[i]=matriz[i][27];
129     }
130     //Formato de los datos
131     double l29[N];
132     double l30[N];
133     for(int i=0;i<N;i++){
134         l29[i]=matriz[i][28];
135         l30[i]=matriz[i][29];
136     }
137     //Formato de los datos
138     double l31[N];
139     double l32[N];
140     for(int i=0;i<N;i++){
141         l31[i]=matriz[i][30];
142         l32[i]=matriz[i][31];
143     }
144     //Formato de los datos
145     double l33[N];
146     double l34[N];
147     for(int i=0;i<N;i++){
148         l33[i]=matriz[i][32];
149         l34[i]=matriz[i][33];
150     }
151     //Formato de los datos
152     double l35[N];
153     double l36[N];
154     for(int i=0;i<N;i++){
155         l35[i]=matriz[i][34];
156         l36[i]=matriz[i][35];
157     }
158     //Formato de los datos
159     double l37[N];
160     double l38[N];
161     for(int i=0;i<N;i++){
162         l37[i]=matriz[i][36];
163         l38[i]=matriz[i][37];
164     }
165     //Formato de los datos
166     double l39[N];
167     double l40[N];
168     for(int i=0;i<N;i++){
169         l39[i]=matriz[i][38];
170         l40[i]=matriz[i][39];
171     }
172     //Formato de los datos
173     double l41[N];
174     double l42[N];
175     for(int i=0;i<N;i++){
176         l41[i]=matriz[i][40];
177         l42[i]=matriz[i][41];
178     }
179     //Formato de los datos
180     double l43[N];
181     double l44[N];
182     for(int i=0;i<N;i++){
183         l43[i]=matriz[i][42];
184         l44[i]=matriz[i][43];
185     }
186     //Formato de los datos
187     double l45[N];
188     double l46[N];
189     for(int i=0;i<N;i++){
190         l45[i]=matriz[i][44];
191         l46[i]=matriz[i][45];
192     }
193     //Formato de los datos
194     double l47[N];
195     double l48[N];
196     for(int i=0;i<N;i++){
197         l47[i]=matriz[i][46];
198         l48[i]=matriz[i][47];
199     }
200     //Formato de los datos
201     double l49[N];
202     double l50[N];
203     for(int i=0;i<N;i++){
204         l49[i]=matriz[i][48];
205         l50[i]=matriz[i][49];
206     }
207     //Formato de los datos
208     double l51[N];
209     double l52[N];
210     for(int i=0;i<N;i++){
211         l51[i]=matriz[i][50];
212         l52[i]=matriz[i][51];
213     }
214     //Formato de los datos
215     double l53[N];
216     double l54[N];
217     for(int i=0;i<N;i++){
218         l53[i]=matriz[i][52];
219         l54[i]=matriz[i][53];
220     }
221     //Formato de los datos
222     double l55[N];
223     double l56[N];
224     for(int i=0;i<N;i++){
225         l55[i]=matriz[i][54];
226         l56[i]=matriz[i][55];
227     }
228     //Formato de los datos
229     double l57[N];
230     double l58[N];
231     for(int i=0;i<N;i++){
232         l57[i]=matriz[i][56];
233         l58[i]=matriz[i][57];
234     }
235     //Formato de los datos
236     double l59[N];
237     double l60[N];
238     for(int i=0;i<N;i++){
239         l59[i]=matriz[i][58];
240         l60[i]=matriz[i][59];
241     }
242     //Formato de los datos
243     double l61[N];
244     double l62[N];
245     for(int i=0;i<N;i++){
246         l61[i]=matriz[i][60];
247         l62[i]=matriz[i][61];
248     }
249     //Formato de los datos
250     double l63[N];
251     double l64[N];
252     for(int i=0;i<N;i++){
253         l63[i]=matriz[i][62];
254         l64[i]=matriz[i][63];
255     }
256     //Formato de los datos
257     double l65[N];
258     double l66[N];
259     for(int i=0;i<N;i++){
260         l65[i]=matriz[i][64];
261         l66[i]=matriz[i][65];
262     }
263     //Formato de los datos
264     double l67[N];
265     double l68[N];
266     for(int i=0;i<N;i++){
267         l67[i]=matriz[i][66];
268         l68[i]=matriz[i][67];
269     }
270     //Formato de los datos
271     double l69[N];
272     double l70[N];
273     for(int i=0;i<N;i++){
274         l69[i]=matriz[i][68];
275         l70[i]=matriz[i][69];
276     }
277     //Formato de los datos
278     double l71[N];
279     double l72[N];
280     for(int i=0;i<N;i++){
281         l71[i]=matriz[i][70];
282         l72[i]=matriz[i][71];
283     }
284     //Formato de los datos
285     double l73[N];
286     double l74[N];
287     for(int i=0;i<N;i++){
288         l73[i]=matriz[i][72];
289         l74[i]=matriz[i][73];
290     }
291     //Formato de los datos
292     double l75[N];
293     double l76[N];
294     for(int i=0;i<N;i++){
295         l75[i]=matriz[i][74];
296         l76[i]=matriz[i][75];
297     }
298     //Formato de los datos
299     double l77[N];
300     double l78[N];
301     for(int i=0;i<N;i++){
302         l77[i]=matriz[i][76];
303         l78[i]=matriz[i][77];
304     }
305     //Formato de los datos
306     double l79[N];
307     double l80[N];
308     for(int i=0;i<N;i++){
309         l79[i]=matriz[i][78];
310         l80[i]=matriz[i][79];
311     }
312     //Formato de los datos
313     double l81[N];
314     double l82[N];
315     for(int i=0;i<N;i++){
316         l81[i]=matriz[i][80];
317         l82[i]=matriz[i][81];
318     }
319     //Formato de los datos
320     double l83[N];
321     double l84[N];
322     for(int i=0;i<N;i++){
323         l83[i]=matriz[i][82];
324         l84[i]=matriz[i][83];
325     }
326     //Formato de los datos
327     double l85[N];
328     double l86[N];
329     for(int i=0;i<N;i++){
330         l85[i]=matriz[i][84];
331         l86[i]=matriz[i][85];
332     }
333     //Formato de los datos
334     double l87[N];
335     double l88[N];
336     for(int i=0;i<N;i++){
337         l87[i]=matriz[i][86];
338         l88[i]=matriz[i][87];
339     }
340     //Formato de los datos
341     double l89[N];
342     double l90[N];
343     for(int i=0;i<N;i++){
344         l89[i]=matriz[i][88];
345         l90[i]=matriz[i][89];
346     }
347     //Formato de los datos
348     double l91[N];
349     double l92[N];
350     for(int i=0;i<N;i++){
351         l91[i]=matriz[i][90];
352         l92[i]=matriz[i][91];
353     }
354     //Formato de los datos
355     double l93[N];
356     double l94[N];
357     for(int i=0;i<N;i++){
358         l93[i]=matriz[i][92];
359         l94[i]=matriz[i][93];
360     }
361     //Formato de los datos
362     double l95[N];
363     double l96[N];
364     for(int i=0;i<N;i++){
365         l95[i]=matriz[i][94];
366         l96[i]=matriz[i][95];
367     }
368     //Formato de los datos
369     double l97[N];
370     double l98[N];
371     for(int i=0;i<N;i++){
372         l97[i]=matriz[i][96];
373         l98[i]=matriz[i][97];
374     }
375     //Formato de los datos
376     double l99[N];
377     double l100[N];
378     for(int i=0;i<N;i++){
379         l99[i]=matriz[i][98];
380         l100[i]=matriz[i][99];
381     }
382     //Formato de los datos
383     double l101[N];
384     double l102[N];
385     for(int i=0;i<N;i++){
386         l101[i]=matriz[i][100];
387         l102[i]=matriz[i][101];
388     }
389     //Formato de los datos
390     double l103[N];
391     double l104[N];
392     for(int i=0;i<N;i++){
393         l103[i]=matriz[i][102];
394         l104[i]=matriz[i][103];
395     }
396     //Formato de los datos
397     double l105[N];
398     double l106[N];
399     for(int i=0;i<N;i++){
400         l105[i]=matriz[i][104];
401         l106[i]=matriz[i][105];
402     }
403     //Formato de los datos
404     double l107[N];
405     double l108[N];
406     for(int i=0;i<N;i++){
407         l107[i]=matriz[i][106];
408         l108[i]=matriz[i][107];
409     }
410     //Formato de los datos
411     double l109[N];
412     double l110[N];
413     for(int i=0;i<N;i++){
414         l109[i]=matriz[i][108];
415         l110[i]=matriz[i][109];
416     }
417     //Formato de los datos
418     double l111[N];
419     double l112[N];
420     for(int i=0;i<N;i++){
421         l111[i]=matriz[i][110];
422         l112[i]=matriz[i][111];
423     }
424     //Formato de los datos
425     double l113[N];
426     double l114[N];
427     for(int i=0;i<N;i++){
428         l113[i]=matriz[i][112];
429         l114[i]=matriz[i][113];
430     }
431     //Formato de los datos
432     double l115[N];
433     double l116[N];
434     for(int i=0;i<N;i++){
435         l115[i]=matriz[i][114];
436         l116[i]=matriz[i][115];
437     }
438     //Formato de los datos
439     double l117[N];
440     double l118[N];
441     for(int i=0;i<N;i++){
442         l117[i]=matriz[i][116];
443         l118[i]=matriz[i][117];
444     }
445     //Formato de los datos
446     double l119[N];
447     double l120[N];
448     for(int i=0;i<N;i++){
449         l119[i]=matriz[i][118];
450         l120[i]=matriz[i][119];
451     }
452     //Formato de los datos
453     double l121[N];
454     double l122[N];
455     for(int i=0;i<N;i++){
456         l121[i]=matriz[i][120];
457         l122[i]=matriz[i][121];
458     }
459     //Formato de los datos
460     double l123[N];
461     double l124[N];
462     for(int i=0;i<N;i++){
463         l123[i]=matriz[i][122];
464         l124[i]=matriz[i][123];
465     }
466     //Formato de los datos
467     double l125[N];
468     double l126[N];
469     for(int i=0;i<N;i++){
470         l125[i]=matriz[i][124];
471         l126[i]=matriz[i][125];
472     }
473     //Formato de los datos
474     double l127[N];
475     double l128[N];
476     for(int i=0;i<N;i++){
477         l127[i]=matriz[i][126];
478         l128[i]=matriz[i][127];
479     }
480     //Formato de los datos
481     double l129[N];
482     double l130[N];
483     for(int i=0;i<N;i++){
484         l129[i]=matriz[i][128];
485         l130[i]=matriz[i][129];
486     }
487     //Formato de los datos
488     double l131[N];
489     double l132[N];
490     for(int i=0;i<N;i++){
491         l131[i]=matriz[i][130];
492         l132[i]=matriz[i][131];
493     }
494     //Formato de los datos
495     double l133[N];
496     double l134[N];
497     for(int i=0;i<N;i++){
498         l133[i]=matriz[i][132];
499         l134[i]=matriz[i][133];
500     }
501     //Formato de los datos
502     double l135[N];
503     double l136[N];
504     for(int i=0;i<N;i++){
505         l135[i]=matriz[i][134];
506         l136[i]=matriz[i][135];
507     }
508     //Formato de los datos
509     double l137[N];
510     double l138[N];
511     for(int i=0;i<N;i++){
512         l137[i]=matriz[i][136];
513         l138[i]=matriz[i][137];
514     }
515     //Formato de los datos
516     double l139[N];
517     double l140[N];
518     for(int i=0;i<N;i++){
519         l139[i]=matriz[i][138];
520         l140[i]=matriz[i][139];
521     }
522     //Formato de los datos
523     double l141[N];
524     double l142[N];
525     for(int i=0;i<N;i++){
526         l141[i]=matriz[i][140];
527         l142[i]=matriz[i][141];
528     }
529     //Formato de los datos
530     double l143[N];
531     double l144[N];
532     for(int i=0;i<N;i++){
533         l143[i]=matriz[i][142];
534         l144[i]=matriz[i][143];
535     }
536     //Formato de los datos
537     double l145[N];
538     double l146[N];
539     for(int i=0;i<N;i++){
540         l145[i]=matriz[i][144];
541         l146[i]=matriz[i][145];
542     }
543     //Formato de los datos
544     double l147[N];
545     double l148[N];
546     for(int i=0;i<N;i++){
547         l147[i]=matriz[i][146];
548         l148[i]=matriz[i][147];
549     }
550     //Formato de los datos
551     double l149[N];
552     double l150[N];
553     for(int i=0;i<N;i++){
554         l149[i]=matriz[i][148];
555         l150[i]=matriz[i][149];
556     }
557     //Formato de los datos
558     double l151[N];
559     double l152[N];
560     for(int i=0;i<N;i++){
561         l151[i]=matriz[i][150];
562         l152[i]=matriz[i][151];
563     }
564     //Formato de los datos
565     double l153[N];
566     double l154[N];
567     for(int i=0;i<N;i++){
568         l153[i]=matriz[i][152];
569         l154[i]=matriz[i][153];
570     }
571     //Formato de los datos
572     double l155[N];
573     double l156[N];
574     for(int i=0;i<N;i++){
575         l155[i]=matriz[i][154];
576         l156[i]=matriz[i][155];
577     }
578     //Formato de los datos
579     double l157[N];
580     double l158[N];
581     for(int i=0;i<N;i++){
582         l157[i]=matriz[i][156];
583         l158[i]=matriz[i][157];
584     }
585     //Formato de los datos
586     double l159[N];
587     double l160[N];
588     for(int i=0;i<N;i++){
589         l159[i]=matriz[i][158];
590         l160[i]=matriz[i][159];
591     }
592     //Formato de los datos
593     double l161[N];
594     double l162[N];
595     for(int i=0;i<N;i++){
596         l161[i]=matriz[i][160];
597         l162[i]=matriz[i][161];
598     }
599     //Formato de los datos
600     double l163[N];
601     double l164[N];
602     for(int i=0;i<N;i++){
603         l163[i]=matriz[i][162];
604         l164[i]=matriz[i][163];
605     }
606     //Formato de los datos
607     double l165[N];
608     double l166[N];
609     for(int i=0;i<N;i++){
610         l165[i]=matriz[i][164];
611         l166[i]=matriz[i][165];
612     }
613     //Formato de los datos
614     double l167[N];
615     double l168[N];
616     for(int i=0;i<N;i++){
617         l167[i]=matriz[i][166];
618         l168[i]=matriz[i][167];
619     }
620     //Formato de los datos
621     double l169[N];
622     double l170[N];
623     for(int i=0;i<N;i++){
624         l169[i]=matriz[i][168];
625         l170[i]=matriz[i][169];
626     }
627     //Formato de los datos
628     double l171[N];
629     double l172[N];
630     for(int i=0;i<N;i++){
631         l171[i]=matriz[i][170];
632         l172[i]=matriz[i][171];
633     }
634     //Formato de los datos
635     double l173[N];
636     double l174[N];
637     for(int i=0;i<N;i++){
638         l173[i]=matriz[i][172];
639         l174[i]=matriz[i][173];
640     }
641     //Formato de los datos
642     double l175[N];
643     double l176[N];
644     for(int i=0;i<N;i++){
645         l175[i]=matriz[i][174];
646         l176[i]=matriz[i][175];
647     }
648     //Formato de los datos
649     double l177[N];
650     double l178[N];
651     for(int i=0;i<N;i++){
652         l177[i]=matriz[i][176];
653         l178[i]=matriz[i][177];
654     }
655     //Formato de los datos
656     double l179[N];
657     double l180[N];
658     for(int i=0;i<N;i++){
659         l179[i]=matriz[i][178];
660         l180[i]=matriz[i][179];
661     }
662     //Formato de los datos
663     double l181[N];
664     double l182[N];
665     for(int i=0;i<N;i++){
666         l181[i]=matriz[i][180];
667         l182[i]=matriz[i][181];
668     }
669     //Formato de los datos
670     double l183[N];
671     double l184[N];
672     for(int i=0;i<N;i++){
673         l183[i]=matriz[i][182];
674         l184[i]=matriz[i][183];
675     }
676     //Formato de los datos
677     double l185[N];
678     double l186[N];
679     for(int i=0;i<N;i++){
680         l185[i]=matriz[i][184];
681         l186[i]=matriz[i][185];
682     }
683     //Formato de los datos
684     double l187[N];
685     double l188[N];
686     for(int i=0;i<N;i++){
687         l187[i]=matriz[i][186];
688         l188[i]=matriz[i][187];
689     }
690     //Formato de los datos
691     double l189[N];
692     double l190[N];
693     for(int i=0;i<N;i++){
694         l189[i]=matriz[i][188];
695         l190[i]=matriz[i][189];
696     }
697     //Formato de los datos
698     double l191[N];
699     double l192[N];
700     for(int i=0;i<N;i++){
701         l191[i]=matriz[i][190];
702         l192[i]=matriz[i][191];
703     }
704     //Formato de los datos
705     double l193[N];
706     double l194[N];
707     for(int i=0;i<N;i++){
708         l193[i]=matriz[i][192];
709         l194[i]=matriz[i][193];
710     }
711     //Formato de los datos
712     double l195[N];
713     double l196[N];
714     for(int i=0;i<N;i++){
715         l195[i]=matriz[i][194];
716         l196[i]=matriz[i][195];
717     }
718     //Formato de los datos
719     double l197[N];
720     double l198[N];
721     for(int i=0;i<N;i++){
722         l197[i]=matriz[i][196];
723         l198[i]=matriz[i][197];
724     }
725     //Formato de los datos
726     double l199[N];
727     double l200[N];
728     for(int i=0;i<N;i++){
729         l199[i]=matriz[i][198];
730         l200[i]=matriz[i][199];
731     }
732     //Formato de los datos
733     double l201[N];
734     double l202[N];
735     for(int i=0;i<N;i++){
736         l201[i]=matriz[i][200];
737         l202[i]=matriz[i][201];
738     }
739     //Formato de los datos
740     double l203[N];
741     double l204[N];
742     for(int i=0;i<N;i++){
743         l203[i]=matriz[i][202];
744         l204[i]=matriz[i][203];
745     }
746     //Formato de los datos
747     double l205[N];
748     double l206[N];
749     for(int i=0;i<N;i++){
750         l205[i]=matriz[i][204];
751         l206[i]=matriz[i][205];
752     }
753     //Formato de los datos
754     double l207[N];
755     double l208[N];
756     for(int i=0;i<N;i++){
757         l207[i]=matriz[i][206];
758         l208[i]=matriz[i][207];
759     }
760     //Formato de los datos
761     double l209[N];
762     double l210[N];
763     for(int i=0;i<N;i++){
764         l209[i]=matriz[i][208];
765         l210[i]=matriz[i][209];
766     }
767     //Formato de los datos
768     double l211[N];
769     double l212[N];
770     for(int i=0;i<N;i++){
771         l211[i]=matriz[i][210];
772         l212[i]=matriz[i][211];
773     }
774     //Formato de los datos
775     double l213[N];
776     double l214[N];
777     for(int i=0;i<N;i++){
778         l213[i]=matriz[i][212];
779         l214[i]=matriz[i][213];
780     }
781     //Formato de los datos
782     double l215[N];
783     double l216[N];
784     for(int i=0;i<N;i++){
785         l215[i]=matriz[i][214];
786         l216[i]=matriz[i][215];
787     }
788     //Formato de los datos
789     double l217[N];
790     double l218[N];
791     for(int i=0;i<N;i++){
792         l217[i]=matriz[i][216];
793         l218[i]=matriz[i][217];
794     }
795     //Formato de los datos
796     double l219[N];
797     double l220[N];
798     for(int i=0;i<N;i++){
799         l219[i]=matriz[i][218];
800         l220[i]=matriz[i][219];
801     }
802     //Formato de los datos
803     double l221[N];
804     double l222[N];
805     for(int i=0;i<N;i++){
806         l221[i]=matriz[i][220];
807         l222[i]=matriz[i][221];
808     }
809     //Formato de los datos
810     double l223[N];
811     double l224[N];
812     for(int i=0;i<N;i++){
813         l223[i]=matriz[i][222];
814         l224[i]=matriz[i][223];
815     }
816     //Formato de los datos
817     double l225[N];
818     double l226[N];
819     for(int i=0;i<N;i++){
820         l225[i]=matriz[i][224];
821         l226[i]=matriz[i][225];
822     }
823     //Formato de los datos
824     double l227[N];
825     double l228[N];
826     for(int i=0;i<N;i++){
827         l227[i]=matriz[i][226];
828         l228[i]=matriz[i][227];
829     }
830     //Formato de los datos
831     double l229[N];
832     double l230[N];
833     for(int i=0;i<N;i++){
834         l229[i]=matriz[i][228];
835         l230[i]=matriz[i][229];
836     }
837     //Formato de los datos
838     double l231[N];
839     double l232[N];
840     for(int i=0;i<N;i++){
841         l231[i]=matriz[i][230];
842         l232[i]=matriz[i][231];
843     }
844     //Formato de los datos
845     double l233[N];
846     double l234[N];
847     for(int i=0;i<N;i++){
848         l233[i]=matriz[i][232];
849         l234[i]=matriz[i][233];
850     }
851     //Formato de los datos
852     double l235[N];
853     double l236[N];
854     for(int i=0;i<N;i++){
855         l235[i]=matriz[i][234];
856         l236[i]=matriz[i][235];
857     }
858     //Formato de los datos
859     double l237[N];
860     double l238[N];
861     for(int i=0;i<N;i++){
862         l237[i]=matriz[i][236];
863         l238[i]=matriz[i][237];
864     }
865     //Formato de los datos
866     double l239[N];
867     double l240[N];
868     for(int i=0;i<N;i++){
869         l239[i]=matriz[i][238];
870         l240[i]=matriz[i][239];
871     }
872     //Formato de los datos
873     double l241[N];
874     double l242[N];
875     for(int i=0;i<N;i++){
876         l241[i]=matriz[i][240];
877         l242[i]=matriz[i][241];
878     }
879     //Formato de los datos
880     double l243[N];
881     double l244[N];
882     for(int i=0;i<N;i++){
883         l243[i]=matriz[i][242];
884         l244[i]=matriz[i][243];
885     }
886     //Formato de los datos
887     double l245[N];
888     double l246[N];
889     for(int i=0;i<N;i++){
890         l245[i]=matriz[i][244];
891         l246[i]=matriz[i][245];
892     }
893     //Formato de los datos
894     double l247[N];
895     double l248[N];
896     for(int i=0;i<N;i++){
897         l247[i]=matriz[i][246];
898         l248[i]=matriz[i][247];
899     }
900     //Formato de los datos
901     double l249[N];
902     double l250[N];
903     for(int i=0;i<N;i++){
904         l249[i]=matriz[i][248];
905         l250[i]=matriz[i][249];
906     }
907     //Formato de los datos
908     double l251[N];
909     double l252[N];
910     for(int i=0;i<N;i++){
911         l251[i]=matriz[i][250];
912         l252[i]=matriz[i][251];
913     }
914     //Formato de los datos
915     double l253[N];
916     double l254[N];
917     for(int i=
```

J.M.T. Thompson and H.B. Stewart, Nonlinear Dynamics and Chaos (2nd edition), 2002.

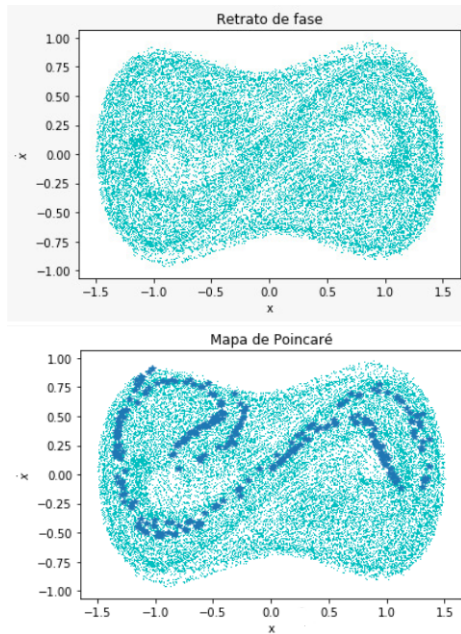


Figura. 5: Retrato de fase y Mapa de Poincaré del sistema para los parámetros fijados

IV. CONCLUSIONES

Se logró solucionar mediante la implementación en C++ el sistema de Duffing para unos valores determinados de los parámetros, en los que se presenta caos. Logrando construir satisfactoriamente tanto el retrato de fase a partir de la integración y el mapa de Poincaré a partir de su construcción, dando cuenta así del caos que el sistema presenta para este caso en particular. La graficación se hizo a partir de la librería Matplotlib de Python dado que supera en calidad a las que se pueden lograr con las herramientas disponibles en C++.

V. REFERENCIAS

- Strogatz, S.H, Nonlinear dynamics and chaos. 1994
Bronson, G. C++ para ingeniería y ciencias. 2006
Burde, R. Numerical Analysis, 9th edition. 2011