Universidad de Guadalajara

centro universitario de ciencias exactas e ingenierías

Tarea 3: SDYC

Luis Ramón Mendoza Escareño

Licenciatura en Física | Sistemas Dinámicos y Caos

07 de Marzo del 2023

I. Resumen

En la práctica se calculó analíticamente los puntos de intersección de una trayectoria con movimiento rectilíneo uniforme y ciertas fronteras. Además de adaptar el código del mapeo hacia la superficie de Poincaré y se tomaron dos casos específicos $q_1 = q_2$,$p_1 = p_2$ y $q_1 = 0$,$p_1 = 0$ pertenecientes a trayectorias 'fundamentales' las cuales se calcularon después de ciertas perturbaciones.

Pd: La parte del cálculo analítico lo dejaremos hasta el final para primero generar el documento en látex.

II.

I. a)

Para hacer que se pudieran tomar varios puntos sólo se generó la siguiente función:

```
  def extra_points(vec_2):
a=[]
for i in range(len(vec_2)):
  a.append(Poin_care(vec_2[i]))
return a
```

En la cual sólo se tienen que poner los puntos en el mapa de poincaré de la siguiente forma: $[[\vec{p_1}, \vec{p_2}, ..., \vec{p_n}]]$ y regresa la misma lista anidando más listas dentro con cada uno de los elementos mapeado al siguiente punto en la superficie de Poincaré.
Nota: Poin_care() es la función que definimos en la tarea pasada, la cual por cierto ya arreglamos.

II. b)

Para este inciso, ya que necesitamos dar una pequeña perturbación a un punto, decidimos generar puntos aleatorios alrededor de esta, con la siguiente función:

```
def aleat(lis,num,bot,top):
 pp_1 = np.array([lis] * num, dtype='float64')
 dp = np.random.uniform(bot, top, num)
 dq = np.random.uniform(bot, top, num)

 pp_1[:, 0] += dp
 pp_1[:, 1] += dq

 return list(pp_1)
```

Que dándole una lista con el primer punto, nos genera una lista con 'num' puntos alrededor de esta, (se genera como una lista ya que así es el formato que se debe tener para ser compatible con la función de varios puntos que generamos en (a)), podemos dar el rango de dispersión con top y bot.

Comenzando por la trayectoria fundamental $q_1 = p_1 = 0$ el movimiento armónico simple en q_2, llamamos a las funciones de arriba de dispersión en (0,0) con dispersiones de ± 0.5 y llamamos a la función extra_points.

```
po_1=[0,0,0]
po_1=aleat(po_1,100,-0.5,0.5)

points_f=np.array(extra_points(po_1))

po_1=np.array(po_1)
```

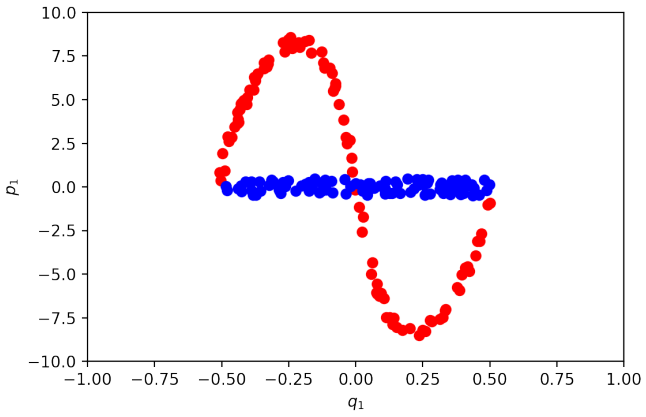

Figura 1: Mapa de Poincaré para puntos con dispersión de ± 0.5 alrededor del origen, se generaron 100 puntos. Los puntos rojos son los mapeos de Poincaré a cada uno de estos. Nota: los límites del plot en x e y son desiguales.

Para el segundo mapa con $q_1 = q_2$, $p_1 = p_2$, de manera similar se generó una dispersión de 0.5 y 100 puntos en el punto $(0, \sqrt{mE})$ y con las funciones antes mencionadas se hizo el mapa.

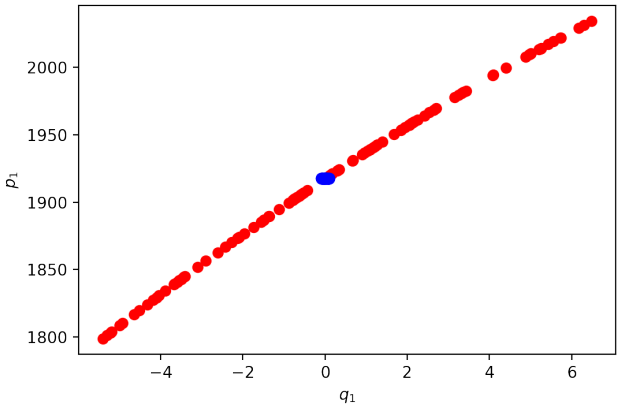

Figura 2: Mapa de Poincaré para puntos con dispersión de ± 0.5 alrededor del origen (puntos azules), se generaron 100 puntos. Los puntos rojos son los mapeos de Poincaré a cada uno de estos. Nota: los límites del plot en x e y son desiguales.

De igual manera, queda el acceso al código en google colab.