
Bitcora de la tesis

Pregrado de Física

Santiago Pea Martnez

spenam@unal.edu.co

Inicio 7 Febrero 2019

Índice general

Jueves, 7 Febrero 2019	1
1 Inicio de la carrera contra reloj	1
Martes, 12 Febrero 2019	2
1 Ecuaciones de Hamilton del sistema y desarrollo para pequenas oscilaciones	2
2 Cdigos para las ecuaciones de Hamilton de para 1 oscilador acoplado . . .	4
3 Cdigo para el integrador de Ruth de 4to orden	4
4 Cdigo para integrador de Yoshida de 6to orden	5
5 Resumen y lo que falta por hacer de hoy	6
6 This is another example experiment	6
7 Rpidamente como recuperar los archivos que estaba copiando en el ssh . .	7
Sbado, 16 Febrero 2019	8
1 Cdigo de Ruth de 3er orden	8
2 ms dilemas con los cdigos!!!!	8
3 Resumen y lo que falta por hacer de hoy v2	12
Lunes, 25 Febrero 2019	13
1 Secciones de poincar para $m=1a=0.25b=0.01M=0.1w=0.8g=0.1$ y condiciones iniciales bolita quieta y oscilador $P=X=0.1$	13
Domingo, 17 Marzo 2019	14
1 Secciones de poincar ya que las anteriores no sirvieron	14
Martes, 19 Marzo 2019	16
1 Comandos para conectarse por ssh	16
Sabado, 13 Abril 2019	17
1 Parece que el integrador funciona pero a la ves no funciona	17
Viernes, 19 Abril 2019	23
1 Algoritmo terminado para los integradores	23
Lunes, 29 Abril 2019	24
1 Papers de nuevos integradores symplecticos que thomas me paso	24
2 This is yet another example experiment	26

Índice general

Friday, 26 March 2010	28
1 This shows a sample table	28
Saturday, 27 March 2010	29
1 Bulleted list example	29
2 This is an example experiment	29
3 This is another example experiment	29
Bibliography	32

Jueves, 7 Febrero 2019

1. Inicio de la carrera contra reloj

Hoy en la tarde estuve hablando con scar, le conté de mis ideas sobre terminar temprano la tesis y acabar la carrera pronto con el fin de irme a Japn. l se mostró muy entusiasmado con mi idea, me dio consejos para posiblemente llevar a cabo esta tarea (incluyendo el de hacer una bitcora). Esta bitcora va a ser un poco narrativa al mismo tiempo de técnica, creo que así podrá expresar mis pensamientos y quizá leerlos en un futuro cuando no tenga nada que hacer, recordar lo que pensaba cuando tenía 20 años y compararlo con lo que estoy haciendo ahora.

Como soy distraído, estaba estudiando para el parcial del martes de Ondas y recordé que tenía que hacer la bitcora, decidí buscar un template en internet y empezarla ya que de lo contrario puede que nunca la empiece.

Hoy no tengo pensado avanzar en cuanto al proyecto debido al parcial que tengo de Ondas, pero este ejercicio práctico me sirve de alguna manera para llevar cuenta de qué es lo que hago, qué casos tomo, cómo hago mis códigos e incluso como práctica para usar latex así estoy usando solamente una template.

P.D. Estoy escuchando August de Sean Angus Watson, creo que esta canción me ha servido muchas veces para estudiar y relajarme. Vamos a ver cómo nos va!

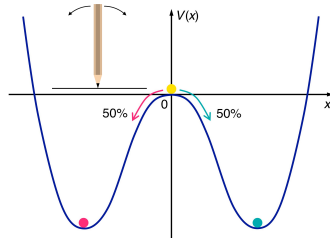
Martes, 12 Febrero 2019

1. Ecuaciones de Hamilton del sistema y desarrollo para pequeñas oscilaciones

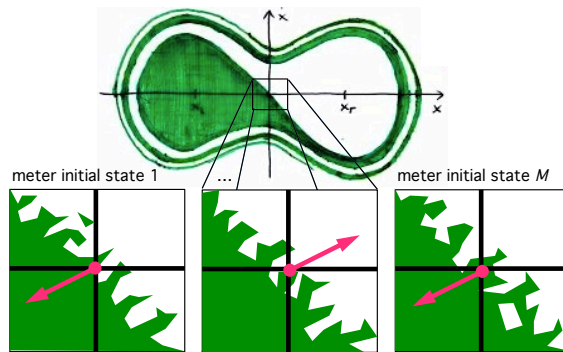
3 Numerical / experimental test

3.2 Classical model

Double well with dissipation and noise



Basins of attraction



$$H(p, x; \mathbf{P}, \mathbf{X}) = H_{dw} + H_c + H_m$$

double well:

$$H_{dw}(p, x) = \frac{p^2}{2m} - a \frac{x^2}{2} + b \frac{x^4}{4}$$

coupling:

$$H_c(p, x; \mathbf{P}, \mathbf{X}) = x \sum_{n=1}^N g_n X_n$$

meter:

$$H_m(\mathbf{P}, \mathbf{X}) = \sum_{n=1}^N \left(\frac{P_n^2}{2M_n} + M_n \omega_n^2 \frac{X_n^2}{2} \right)$$

initial states

double well: meter:

$$\text{"Buridan's ass" equilibrium} \quad p = 0, x = 0 \quad \langle \mathbf{P} \rangle = \mathbf{0}, \langle \mathbf{X} \rangle = \mathbf{0}$$

Figura 1: Hamiltonianos del sistema

Para este sistema se tiene el siguiente hamiltoniano:

$$H(p, x; \mathbf{P}, \mathbf{X}) = H_{dw} + H_c + H_m \quad (0.1)$$

Con el Hamiltoniano del pozo doble (cuartico) dado por:

$$H_{dw} = \frac{p^2}{2m} - a \frac{x^2}{2} + b \frac{x^4}{4} \quad (0.2)$$

El del acoplamiento dado por:

$$H_c(p, x; \mathbf{P}, \mathbf{X}) = x \sum_{n=1}^N g_n X_n \quad (0.3)$$

Y el del medidor:

$$H_m(\mathbf{P}, \mathbf{X}) = \sum_{n=1}^N \left(\frac{P_n^2}{2M_n} + M_n \omega_n^2 \frac{X_n^2}{2} \right) \quad (0.4)$$

Las ecuaciones de Hamilton para el pozo son:

$$\dot{p} = \frac{-\partial H}{\partial x} = ax - bx^3 - \sum_{n=1}^N g_n X_n \quad (0.5)$$

$$\dot{v} = \frac{-1}{m} \frac{\partial H}{\partial x} = \frac{ax}{m} - \frac{bx^3}{m} - \frac{\sum_{n=1}^N g_n X_n}{m} \quad (0.6)$$

$$\dot{x} = \frac{\partial H}{\partial p} = \frac{p}{m} \quad (0.7)$$

Las ecuaciones de Hamilton para los osciladores son:

$$\dot{P}_i = \frac{-\partial H}{\partial X_i} = -xg_i - M_i \omega_i^2 X_i \quad (0.8)$$

$$\dot{V}_i = \frac{-1}{M_i} \frac{\partial H}{\partial X_i} = \frac{-xg_i}{M_i} - \omega_i^2 X_i \quad (0.9)$$

$$\dot{X}_i = \frac{\partial H}{\partial P_i} = \frac{P_i}{M_i} \quad (0.10)$$

Ahora bien, para calcular las frecuencias de pequenas oscilaciones del pozo doble se procede de la siguiente manera. El potencial del pozo est dado por:

$$V(x) = -\frac{1}{2}ax^2 + \frac{1}{4}bx^4 \quad (0.11)$$

Los puntos de equilibrio ocurren cuando el potencial V es mximo o mnimo, es decir:

$$\frac{d}{dx}V = 0 \rightarrow x(bx^2 - a) = 0 \quad (0.12)$$

Las raices de la ecuacin anterior son $x = 0, \pm \sqrt{\frac{a}{b}}$, donde $x = 0$ es un punto de equilibrio inestable y $x = \pm \sqrt{\frac{a}{b}}$ son puntos de equilibrio estables.

Ahora bien, hacemos una expansin de Taylor alrededor del punto de equilibrio estable $x_0 = \sqrt{\frac{a}{b}}$:

$$U(x_0 + \epsilon) = U(x_0) + U'(x_0)\epsilon + \frac{1}{2}U''(x_0)\epsilon^2 + \dots \quad (0.13)$$

El primer trmino es una constante entonces se puede ignorar, el segundo trmino es cero ya que $U'(x_0) = 0$ y el tercer trmino ser $\frac{1}{2}(-a + 3bx^2)|_{x=x_0}$, esto da como resultado:

$$U = \frac{1}{2}(2a)\epsilon^2 \quad (0.14)$$

Es claro que la constante elstica de este potencial es $2a$ (por comparacin con $U = \frac{1}{2}kx^2$). Esto quiere decir que la frecuencia de pequenas oscilaciones en estos puntos de equilibrio estn dadas por:

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{2a}{m}} \quad (0.15)$$

2. Cdigos para las ecuaciones de Hamilton de para 1 oscilador acoplado

```
1 def A(m0, a0, b0, x0, g0, X0): #p punto
2     return a0*x0/m0 - b0*x0*x0*x0/m0 -(g0*X0)/m0
3 def V(m0,p0): #equis punto
4     return p0/m0
5 def Ai(Mi0,wi0,x0,gi0,Xi0): #P punto sub i
6     return -x0*gi0/Mi0-wi0**2.*Xi0
7 def Vi(Mi0,Pi0): #Equis punto sub i
8     return Pi0/Mi0
```

3. Cdigo para el integrador de Ruth de 4to orden

```
1 def ruth4th(xx,pp,mm,XX01,PP01,MM1,ww1,gacople,dt,
2 iteraciones,a_poso,b_poso)#defino funcion con el integrador symplectico
3     Ruth de 4to orden
4 dos=2.*(1./3.) #constante necesaria
5
6 aa=[a,(1./(2.-dos)),(-dos/(2.-dos)),(1./(2.-dos)),0] #constantes de la
7     posicion del integrador
8 bb=[b,(1./(2.*(2.-dos))), (1.-dos)/(2.*(2.-dos)), (1.-dos)/(2.*(2.-dos)),
9     (1./(2.*(2.-dos)))] #constantes de la velocidad del integrador
10
11 x = [xx] #posicion inicial
12 p = [pp] #momento inicial
13 X1 = [XX01] #posicion inicial del primer oscilador
14 P1 = [PP01] #momento inicial del primer oscilador
15 t = [0.] #lista de tiempo
16
17 xi , pi , X1i , P1i= xx, pp , XX01 , PP01
18 for i in range (iteraciones):
19     vini = pi/m
20     vv = V(mm,pi)+bb[1]*A(mm,a_poso,b_poso,xi,gacople,X1i)*dt
21     VV1 = Vi(MM1,P1i)+bb[1]*Ai(MM1,ww1,xi,gacople[1],X1i)*dt
22     xx = xi + aa[1]*vv*dt
```



```

21     XX1 =XX1i + aa[1]*VV1*dt
22     for j in range(2,len(aa)):
23         vv += bb[j]*A(mm,a_poso,b_poso,xx,gacople,XX1)*dt
24     VV1 += bb[j]*Ai(MM1,ww1,xx,gacople[1],XX1)*dt
25     xx += aa[j]*vv*dt
26     XX1 += aa[j]*VV1*dt
27
28     xi = xx
29     pi = m*vv
30     X1i = XX1
31     P1i = MM1*VV1
32
33     x.append(xi)
34     p.append(pi)
35     X1.append(X1i)
36     P1.append(P1i)
37     t.append(t[-1] + dt)

```

4. Codigo para integrador de Yoshida de 6to orden

```

1 #codigo con el integrador de yoshida de sexto orden, los parametros son xx=
  posicion inicial, pp=momento inicial, XX01=posicion inicial del primer
  oscilador, PP01=momento inicial del primer oscilador, MM1=masa del
  primer oscilador, ww1=frecuencia del primer oscilador, gacople= lista
  de los acoples para cada oscilador, dt= paso del integrador,
  iteraciones= numero de iteraciones del integrador, a_poso=parametro a
  del poso, b_poso=parametro b del poso,
2
3 def yoshida(xx,pp,mm,XX01,PP01,MM1,ww1,gacople,dt,iteraciones,a_poso,b_poso
  ): #defino funcion con el integrador symplectico yoshida de 6to orden
4
5
6 aa = [ a, 0.78451361047756, 0.23557321335936, -1.1776799841789,
  1.3151863206839, -1.1776799841789, 0.23557321335936,
  0.78451361047756, 0]
7 bb = [ b, 0.39225680523878, 0.51004341191846, -0.47105338540976,
  0.068753168252520, 0.068753168252520, -0.47105338540976,
  0.51004341191846, 0.39225680523878]
8
9
10 x = [xx] #posicion inicial
11 p = [pp] #momento inicial
12 X1 = [XX01] #posicion inicial del primer oscilador
13 P1 = [PP01] #momento inicial del primer oscilador
14 t = [0.] #lista de tiempo
15
16 xi , pi , X1i , P1i= xx, pp , XX01 , PP01
17 for i in range (iteraciones):
18     vini = pi/m
19     vv = V(mm,pi)+bb[1]*A(mm,a_poso,b_poso,xi,gacople,X1i)*dt
20     VV1 = Vi(MM1,P1i)+bb[1]*Ai(MM1,ww1,xi,gacople[1],X1i)*dt
21     xx = xi + aa[1]*vv*dt
22     XX1 =XX1i + aa[1]*VV1*dt

```

```
23     for j in range(2, len(aa)) :
24         vv += bb[j]*A(mm, a_poso, b_poso, xx, gacople, XX1)*dt
25         VV1 += bb[j]*Ai(MM1, ww1, xx, gacople[1], XX1)*dt
26         xx += aa[j]*vv*dt
27         XX1 += aa[j]*VV1*dt
28
29         xi = xx
30         pi = m*vv
31         X1i = XX1
32         P1i = MM1*VV1
33
34         x.append(xi)
35         p.append(pi)
36         X1.append(X1i)
37         P1.append(P1i)
38         t.append(t[-1] + dt)
39
```

5. Resumen y lo que falta por hacer de hoy

Los desarrollos anteriores fueron hechos con el fin de tener a la mano las ecuaciones tiles a las cuales vuelvo constantemente. Los cdigos por otra parte, son tentativos. Lo que debo hacer en la proxima sesin:

- Comprobar que los cdigos propuestos funcionen, con casos de prueba
- Agregar un cdigo de Ruth de 3er orden
- Una vez que los cdigos funcionen, crear un cdigo para el sistema mio que llame las funciones de los integradores
- Cuando lo anterior sea hecho, agregar los cdigos al repositorio de github

6. This is another example experiment

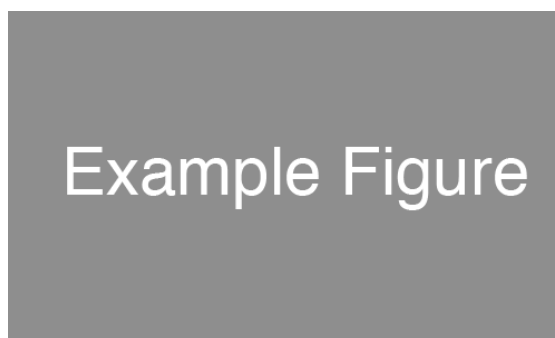


Figura 2: Example figure.

7. Rpidamente como recuperar los archivos que estaba copiando en el ssh

```
1 rsync -P -e ssh spenam@168.176.92.190:mec_estadistica/lambda09.dat /home/  
santiago/Downloads/mec_estadistica/proycod/09
```

Sbado, 16 Febrero 2019

1. Codigo de Ruth de 3er orden

```
1 def ruth3rd (xx,pp,mm,XX01,PP01,MM1,ww1,gacople ,dt ,
2 iteraciones ,a_poso ,b_poso)#defino funcion con el integrador symplectico
   Ruth de 3er orden
3
4 aa=[a,-1./24.,3./4.,7./24.] #constantes de la posicion del integrador
5 bb=[b,1.,-2./3., 2./3.] #constantes de la velocidad del integrador
6
7     x = [xx] #posicion inicial
8     p = [pp] #momento inicial
9     X1 = [XX01] #posicion inicial del primer oscilador
10    P1 = [PP01] #momento inicial del primer oscilador
11    t = [0.] #lista de tiempo
12
13    xi , pi , X1i , P1i= xx, pp , XX01 , PP01
14    for i in range (iteraciones):
15        vini = pi/m
16        vv = V(mm,pi)+bb[1]*A(mm,a_poso ,b_poso ,xi ,gacople ,X1i)*dt
17        VV1 = Vi(MM1,P1i)+bb[1]*Ai(MM1,ww1,xi ,gacople [1] ,X1i)*dt
18        xx = xi + aa[1]*vv*dt
19        XX1 =XX1i + aa[1]*VV1*dt
20        for j in range(2,len(aa)):
21            vv += bb[j]*A(mm,a_poso ,b_poso ,xx ,gacople ,XX1)*dt
22            VV1 += bb[j]*Ai(MM1,ww1,xx ,gacople [1] ,XX1)*dt
23            xx += aa[j]*vv*dt
24            XX1 += aa[j]*VV1*dt
25
26        xi = xx
27        pi = m*vv
28        X1i = XX1
29        P1i = MM1*VV1
30
31        x.append(xi)
32        p.append(pi)
33        X1.append(X1i)
34        P1.append(P1i)
35        t.append(t[-1] + dt)
```

2. ms dilemas con los cdigos!!!!

Hice los cdigos usando los loops, para intentar hacer casos de prueba tom el potencial catico de Henon-Heiles, pero oh sorpresa, no obtena la misma figura. Obtena algo cercano pero no exactamente lo que necesitaba.

En un momento de desesperación, busqué alternativas. Estas posibles alternativas llegaron en Mathematica y voy a probar en Matlab(Octave).

Ocurrió que busqué en internet posibles integradores simplécticos de Mathematica y obtengo resultados un poco distintos para los mismos parámetros a los que ya les había hecho 123081324796 gráficas! Algo bueno es que estas gráficas de Mathematica también dependen aparentemente del paso del integrador.

Estas imágenes muestran lo que quiero decir para los parámetros dados por: $m = 0,1$, $a = 0,25$, $b = 0,01$, $M = 0,01$, $\omega = 2,236067977$, $g = 0,01$ y condiciones iniciales dadas por $P(0) = 0,1$, $X(0) = 0,1$ para un tiempo $t = 300$:

Todo esto usando integradores simplécticos de 4to orden.

- Para $dt = 0,001$:

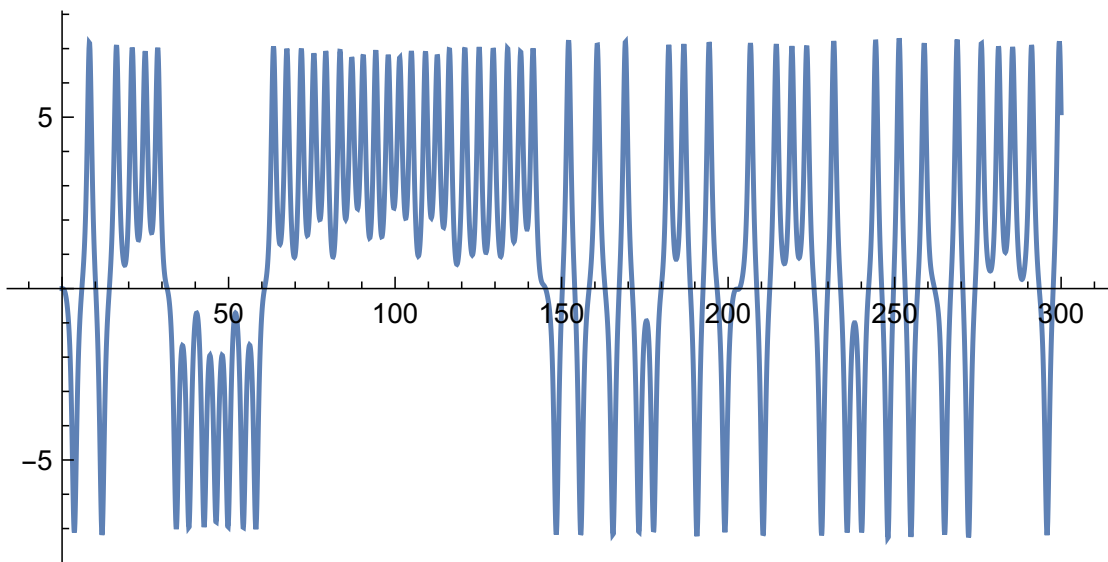


Figura 1: Gráfica x vs t en Mathematica para $dt = 0,001$

posicion bolita vs tiempo para oscilador con condiciones iniciales
PI= 0.1 y XI= 0.1 y energia inicial E= 0.50025
para verlet de 4to orden con dt=e-3

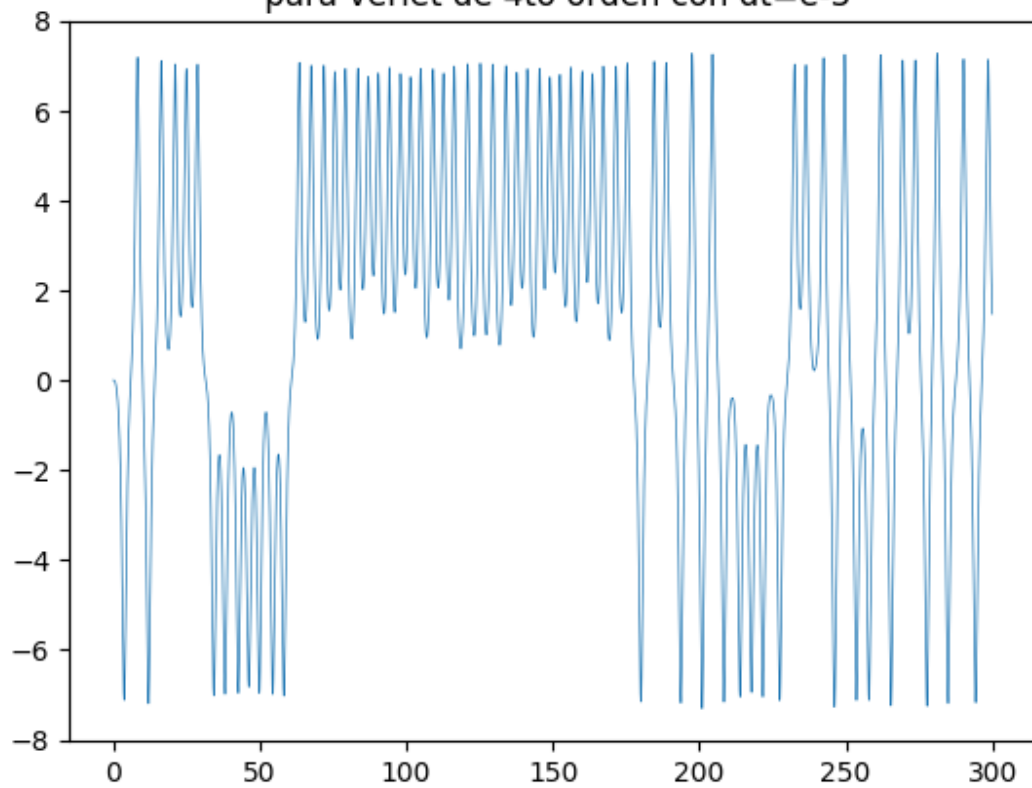


Figura 2: Grfica x vs t en Python para $dt = 0,001$

- Para $dt = 0,0001$:

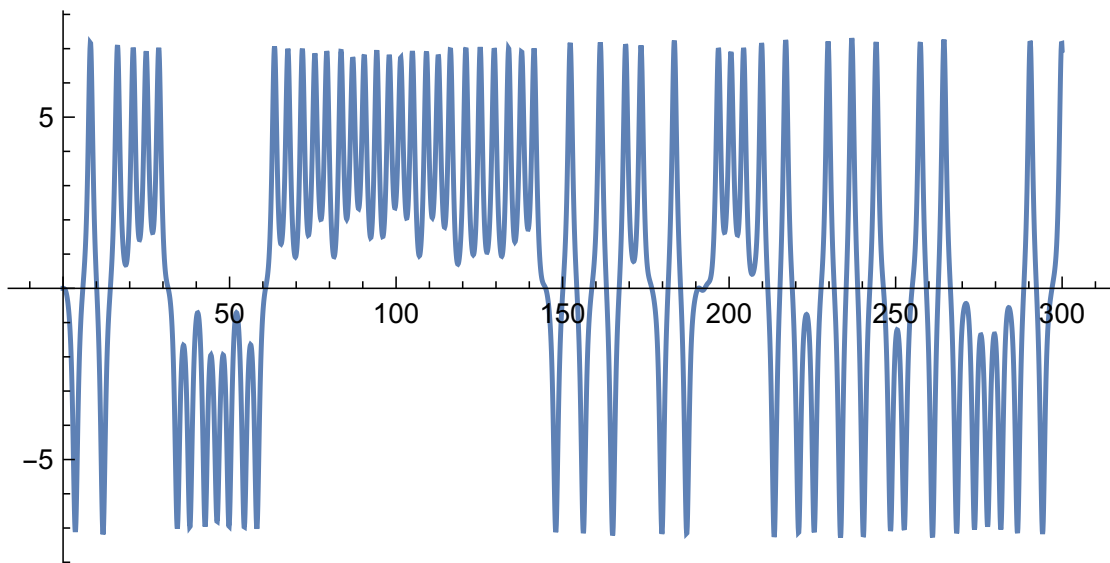


Figura 3: Gráfica x vs t en Mathematica para $dt = 0,0001$

posicion bolita vs tiempo para oscilador con condiciones iniciales
 $PI = 0.1$ y $XI = 0.1$ y energia inicial $E = 0.50025$
para verlet de 4to orden con $dt = e-4$

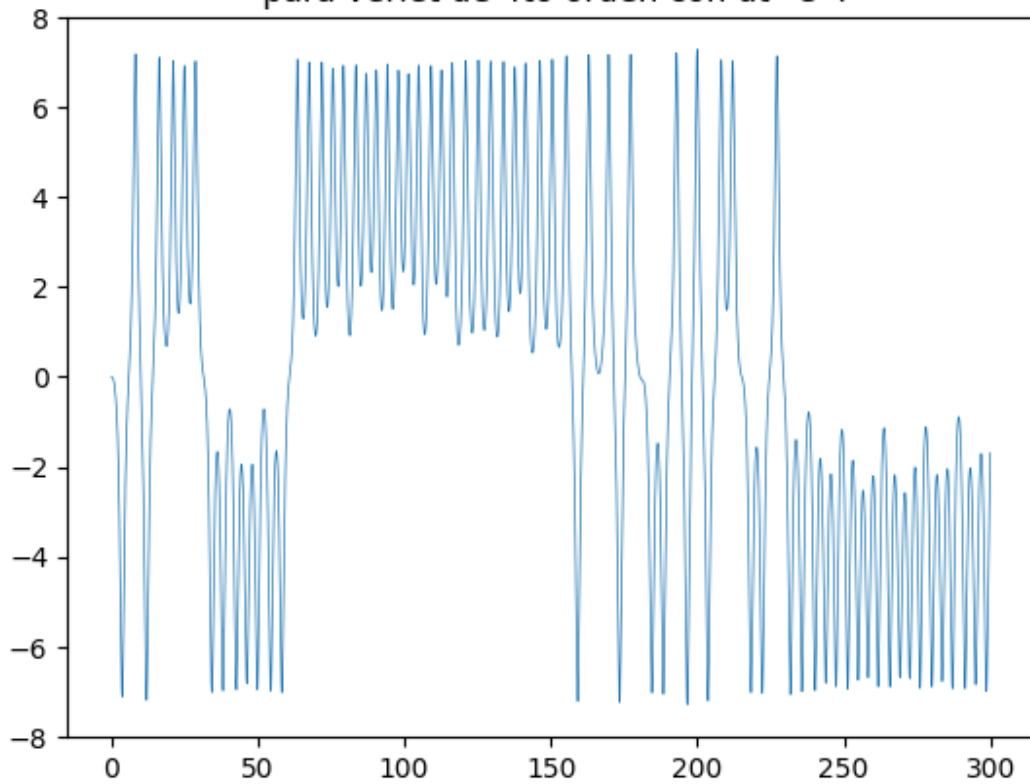


Figura 4: Grfica x vs t en Python para $dt = 0,0001$

3. Resumen y lo que falta por hacer de hoy v2

La bsqueda de los integradores simplcticos que tengan buenos resultados para este sistema empieza a ser cada vez ms complicada. Lo que debo hacer en la prxima sesin otra vez:

- Comprobar que los cdigos propuestos funcionen, con casos de prueba
- Una vez que los cdigos funcionen, crear un cdigo para el sistema mio que llame las funciones de los integradores
- Cuando lo anterior sea hecho, agregar los cdigos al repositorio de github
- Hablar con scar y el profesor para ver que opinan al respecto.

Lunes, 25 Febrero 2019

1. Secciones de poincar para $m=1, a=0.25, b=0.01, M=0.1, w=0.8, g=0.1$ y condiciones iniciales bolita quieta y oscilador $P=X=0.1$

Hice un gif usando 30 secciones de poincar para el mismo problema, las secciones se crean a partir de tomar la frecuencia del oscilador ω aunque ahora que lo pienso, deb haber tomado el periodo del oscilador. Me pregunto si dar lo mismo.

Entonces esta primera parte ser para la frecuencia, la otra parte ser para el periodo:

- gif de seccion de poincar variando un poco el trmino de la frecuencia:

No pude agregar gifs a este archivo jajaja. Entonces me contento con haberle enviado al profesor los gifs de las secciones de poicar para la frecuencia y periodo del oscilador.

Los otros objetivos siguen siendo los mismos:

- Comprobar que los cdigos propuestos funcionen, con casos de prueba
- Una vez que los cdigos funcionen, crear un cdigo para el sistema mio que llame las funciones de los integradores
- Cuando lo anterior sea hecho, agregar los cdigos al repositorio de github
- Hablar con scar y el profesor para ver que opinan al respecto.

Domingo, 17 Marzo 2019

1. Secciones de poincar ya que las anteriores no sirvieron

Dado que las secciones de poincar que intent hacer anteriormente no funcionaron, entonces proced a buscar bien cmo se hacan, en esto encontr un algoritmo que creo que funcionar:

Procedimiento computacional (Algoritmo para el problema de Henon-Heiles):

El procedimiento seguido en este trabajo para calcular las secciones de Poincare en el sistema de Henon-Heiles es el siguiente (se ha empleado el plano $x = 0$ por simplicidad, pero se podria emplear cualquier otro plano).

1. Se parten de unas condiciones iniciales tales que $x(0) = 0$, $y(0) = y_0$, $p_y(0) = p_{y0}$ y una energia $H = H_0$. A partir de estos valores, se calcula $p_x(0)$ a partir de la restriccion $H = H_0$.

$$p_x(0) = \sqrt{2H_0 - p_{y0}^2 - y_0^2 + \frac{2}{3}y_0^3} \quad (0.16)$$

2. Se integra el sistema de ecuaciones diferenciales (4.6) con el integrador symplectico de orden 4 empleado en todo el trabajo.
3. De la integracion de las ecuaciones, el programa almacena los pares (y, p_y) que cumplen que $\epsilon < x < \epsilon$. Siendo ϵ un parametro del programa que indica la precision del mapa de Poincare. Cuanto menor sea ϵ , menos puntos tendra el mapa de Poincare resultante.
4. Una vez se han integrado las ecuaciones y se han almacenado en un archivo todos los pares (y, p_y) para esta condicion inicial, se vuelve al primer paso cambiando las condiciones iniciales con la misma energia y se repite todo el proceso.
5. Repitiendo este procedimiento varias veces se obtiene el mapa de Poincare asociado al sistema dinamico (cuantas mas veces repita, mas rico sera el mapa).

Con esto en mente, el proceso para el hamiltoniano de nuestro problema ser usando la posicin del primer oscilador en $X = 0$:

1. Se parten de unas condiciones iniciales tales que $X(0) = 0$, $p(0) = p_0$, $x(0) = x_0$ y una energia $H = H_0$. A partir de estos valores, se calcula $P(0)$ a partir de la restriccion $H = H_0$.

$$P(0) = \sqrt{2M(H_0 - \frac{p_0^2}{2m} + a\frac{x_0^2}{2} - b\frac{x_0^4}{4})} \quad (0.17)$$

2. Se integra el sistema de ecuaciones diferenciales (4.6) con el integrador symplectico de orden 4 empleado en todo el trabajo.
3. De la integracion de las ecuaciones, el programa almacena los pares (y, p_y) que cumplen que $\epsilon < x < \epsilon$. Siendo ϵ un parametro del programa que indica la precision del mapa de Poincare. Cuanto menor sea ϵ , menos puntos tendra el mapa de Poincare resultante.
4. Una vez se han integrado las ecuaciones y se han almacenado en un archivo todos los pares (y, p_y) para esta condicion inicial, se vuelve al primer paso cambiando las condiciones iniciales con la misma energia y se repite todo el proceso.
5. Repitiendo este procedimiento varias veces se obtiene el mapa de Poincare asociado al sistema dinamico (cuantas mas veces repita, mas rico sera el mapa).

La manera como voy a proceder es haciendo una grilla de 25 puntos al rededor de la zona de equilibrio del oscilador y manteniendo una energia fija, en la primera prueba la energia impuesta es la resultante de las condiciones iniciales del oscilador dadas por $X = 0,1$ y $P = 0,1$.

Los otros objetivos siguen siendo los mismos:

- Comprobar que los cdigos propuestos funcionen, con casos de prueba
- Una vez que los cdigos funcionen, crear un cdigo para el sistema mio que llame las funciones de los integradores
- Cuando lo anterior sea hecho, agregar los cdigos al repositorio de github
- Hablar con scar y el profesor para ver que opinan al respecto.

Martes, 19 Marzo 2019

1. Comandos para conectarse por ssh

ssh spenam@168.176.92.190

la contrasea es el usuario: spenam

desde fuera del ssh copio los archivos que necesito por medio de:

```
1 scp route to file /file userid@168.176.92.190:Path where to save/
```

ejemplo de lo anterior:

```
1 scp ~/Desktop/tesis/poincare/compu_grupo/test_grupo.py spenam@168.176.92.190:poincare/
```

para traer archivos, en este caso fue una carpeta:

```
1 scp -r chaosg@168.176.92.190:poincare ~/Desktop/tesis/poincare/
```

Tambien puedo conectarme al cluster de Delaware por medio de este comando:

```
1 ssh spenam@asterix-login.physics.udel.edu
```

La contrasena es la de todo.

Sabado, 13 Abril 2019

1. Parece que el integrador funciona pero a la ves no funciona

Definitivamente no entiendo esto de los integradores, puesto que decid usar la base del que estaba generalizado, en donde solo tengo que insertar las constantes que definen el integrador. Decid probar las secciones de Poincar del potencial caotico de Henon-Heiles (El cual est ms arriba) y us un integrador de ruth de 3er orden. Luego de jugar con eso y buscar las aparentes condiciones iniciales que funcionaban, el resultado fueron grficas bastante bonitas en donde cre que todo estaba empezando a dar:

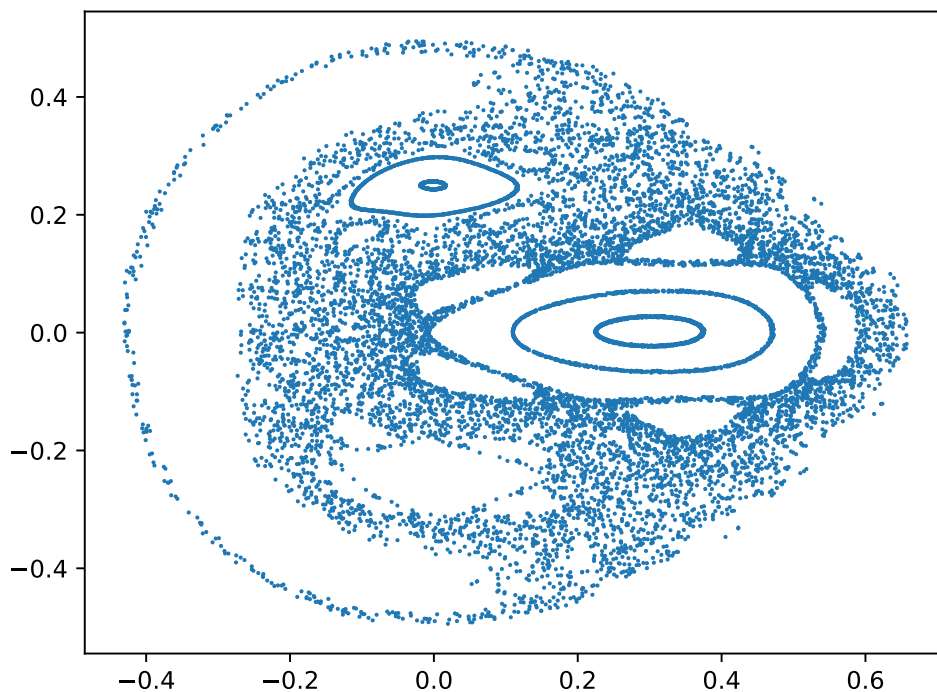


Figura 1: Secciones de Poincar para el potencial de Henon-Heiles

Pareca que lo haba logrado, este es el cdigo que us:

Sabado, 13 Abril 2019

```
1 import math as m
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 def xpunto(vx0): #time derivative of x
6     return vx0
7 def pxpunto(x0,y0,lambda0): #time derivative of px
8     #print (x0,lambda0,y0)
9     return -x0-(2.*lambda0*x0*y0)
10 def ypunto(vy0): #time derivative of y
11     return vy0
12 def pypunto(x0,y0,lambda0): #time derivative of py
13     return -y0-(lambda0*((x0*x0)-(y0*y0)))
14 def energia_poinc(y0,py0,H0): #initial condition for px using the
    constraints
15     print (y0,py0,H0)
16     print ((2.*H0)-(py0*py0)-(y0*y0)+((2./3.)*(y0*y0*y0)))
17     return m.sqrt(abs((2.*H0)-(py0*py0)-(y0*y0)+((2./3.)*(y0*y0*y0))))
18
19
20
21 iteraciones=500000
22 dt=0.01
23 t=[0.]
24 x=[]
25 y=[]
26 px=[]
27 py=[]
28
29 def ruth3rd(xx,yy,ppx,ppy,lambda00):#define function with symplectic
    integrator
30
31     aa=[1,-1./24.,3./4.,7./24.] #constants for position variable of the
    integrator
32     bb=[1,1.,-2./3., 2./3.] #constants of velocity of the integrator
33
34     x.append(xx)
35     y.append(yy)
36     px.append(ppx)
37     py.append(ppy)
38
39
40
41     xi , yi , pxi , pyi= xx, yy, ppx , ppy
42     for i in range (iteraciones):
43         #print (i*1.0*100.0/(iteraciones*1.0))
44         vvx = pxi+bb[1]*pxpunto(xi,yi,lambda00)*dt
45         VVly = pyi+bb[1]*pypunto(xi,yi,lambda00)*dt
46         xx = xi + aa[1]*vvx*dt
47         yy =yi + aa[1]*VVly*dt
48         for j in range(2,len(aa)):
49             vvx += bb[j]*pxpunto(xx,yy,lambda00)*dt
50             VVly += bb[j]*pypunto(xx,yy,lambda00)*dt
51             xx += aa[j]*vvx*dt
```

```

52         yy += aa[j]*VVly*dt
53
54         xi = xx
55         pxi = vvx
56         yi = yy
57         pyi = VVly
58
59         x.append(xi)
60         px.append(pxi)
61         y.append(yi)
62         py.append(pyi)
63         t.append(t[-1] + dt)
64
65 setsrange=4
66 sets=np.array(np.linspace(0., 4., 4),dtype=np.float64)
67 setsl=[1.,2.,3.,4.]
68
69 for I in range(1):
70
71     xpoinc=[]
72     ppoinc=[]
73
74     for ii in range (setsrange):
75         for jj in range (setsrange):
76
77             Hinicial=1./8.
78             x=[0.]
79             y=[-0.15+0.3*(sets[jj]/(4.-1.))]
80             py=[-0.15+0.3*(sets[ii]/(4.-1.))]
81             #print(y[0],py[0],Hinicial)
82             #print(energia_poinc(y[0],py[0],Hinicial))
83             px=[energia_poinc(y[0],py[0],Hinicial)]
84
85             ruth3rd(x[0],y[0],px[0],py[0],1.)
86
87
88     crossings=[]
89     crossingsindex=[]
90     for idx, item in enumerate(x[:-1]): #intersections
91         if x[idx] < 0 and x[idx+1] > 0:
92             crossings.append(x[idx])
93             crossingsindex.append(idx)
94
95
96     for index in crossingsindex:
97         xpoinc.append(y[index])
98         ppoinc.append(py[index])
99
100 np.savetxt('xpoinc.out',xpoinc) #data of the poincare plot
101 np.savetxt('ppoinc.out',ppoinc)

```

Pareca que todo iba viento en popa, pero al intentar hacer lo mismo con un integrador de cuarto orden de ruth no daba lo mismo. No daba lo mismo!!!!!! A pesar de que solo cambi las constantes, el integrador fallaba y daba valores demasiado grandes, no entiendo

Sabado, 13 Abril 2019

porqu:

```
1 import math as m
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 def xpunto(vx0): #time derivative of x
6     return vx0
7 def pxpunto(x0,y0,lambda0): #time derivative of px
8     print (x0,lambda0,y0)
9     return -x0-(2.*lambda0*x0*y0)
10 def ypunto(vy0): #time derivative of y
11     return vy0
12 def pypunto(x0,y0,lambda0): #time derivative of py
13     return -y0-(lambda0*((x0*x0)-(y0*y0)))
14 def energia_poinc(y0,py0,H0): #initial condition for px using the
    constraints
15     print (y0,py0,H0)
16     print ((2.*H0)-(py0*py0)-(y0*y0)+((2./3.)*(y0*y0*y0)))
17     return m.sqrt(abs((2.*H0)-(py0*py0)-(y0*y0)+((2./3.)*(y0*y0*y0))))
18
19
20
21
22 iteraciones=2000
23 dt=0.01
24 t=[0.]
25 x=[]
26 y=[]
27 px=[]
28 py=[]
29
30 def ruth3rd(xx,yy,ppx,ppy,lambda00):#define function with symplectic
    integrator
31
32     dos=np.math.pow(2.0, 1.0/3.0)
33     aa=[1,0.5/(2.0 - dos), 0.5*(1.0-dos)/(2.0 - dos), 0.5*(1.0-dos)/(2.0
        - dos), 0.5/(2.0 - dos)] #constants for position variable of the
    integrator
34     bb=[1,1./ (2.0 - dos),-dos/ (2.0 - dos),1./ (2.0 - dos),0.] #constants
        of velocity of the integrator
35
36     x.append(xx)
37     y.append(yy)
38     px.append(ppx)
39     py.append(ppy)
40
41
42
43     xi , yi , pxi , pyi= xx, yy, ppx , ppy
44     for i in range (iteraciones):
45         #print (i*1.0*100.0/(iteraciones*1.0))
46         vx = pxi+bb[1]*pxpunto(xi,yi,lambda00)*dt
47         Vy = pyi+bb[1]*pypunto(xi,yi,lambda00)*dt
```



```

48     xx = xi + aa[1]*vvx*dt
49     yy = yi + aa[1]*VVly*dt
50     for j in range(2, len(aa)):
51         vvz += bb[j]*pxpunto(xx, yy, lambda00)*dt
52     VVly += bb[j]*pxpunto(xx, yy, lambda00)*dt
53     xx += aa[j]*vvx*dt
54     yy += aa[j]*VVly*dt
55
56     xi = xx
57     pxi = vvz
58     yi = yy
59     pyi = VVly
60
61     x.append(xi)
62     px.append(pxi)
63     y.append(yy)
64     py.append(pyi)
65     t.append(t[-1] + dt)
66
67 setsrange=2
68 sets=np.linspace(0., 4., 4), dtype=np.float64)
69 sets1=[1., 2., 3., 4.]
70
71 for I in range(1):
72
73     xpoinc=[]
74     ppoinc=[]
75
76     for ii in range (setsrange):
77         for jj in range (setsrange):
78
79             Hinicial=1./8.
80             x=[0.]
81             y=[-0.15+0.3*(sets[jj]/(4.-1.))]
82             py=[-0.15+0.3*(sets[ii]/(4.-1.))]
83             #print(y[0], py[0], Hinicial)
84             #print(energia_poinc(y[0], py[0], Hinicial))
85             px=[energia_poinc(y[0], py[0], Hinicial)]
86
87             ruth3rd(x[0], y[0], px[0], py[0], 1.)
88
89
90     crossings=[]
91     crossingsindex=[]
92     for idx, item in enumerate(x[:-1]): #intersections
93         if x[idx] < 0 and x[idx+1] > 0:
94             crossings.append(x[idx])
95             crossingsindex.append(idx)
96
97
98     for index in crossingsindex:
99         xpoinc.append(y[index])
100         ppoinc.append(py[index])
101

```

Sabado, 13 Abril 2019

```
102 np.savetxt('xpoinc4th.out', xpoinc) #data of the poicare plot
103 np.savetxt('ppoinc4th.out', ppoinc)
```

Hice tambien varias pruebas numricas de los integradores para el caso del pndulo y muestra que conserva bien el area, entonces no se que est pasando.

Por fin encontr el error, estaba en el algoritmo del integrador, estaba llamando pxpunto en el paso de evolucion de VV1y!!

Viernes, 19 Abril 2019

1. Algoritmo terminado para los integradores

Luego de que por fin tuviera bien aparentemente los integradores los probé con el sistema y obtuve unos resultados aparentemente satisfactorios, en la carpeta de Poincaré que se llama prueba integrador hice los códigos llamados 3rdsistema y 4th sistema donde supuestamente los tengo en el modo del algoritmo y aparentemente funciona bastante bien. Ahora voy a estudiar los papers que me mandó Thomas donde implementan unos métodos de integradores symplecticos sin constantes negativas donde argumentan que las constantes negativas dan inestabilidad en la integración. Algo interesante es que probé esto con el caso que tiene una frecuencia bastante alta y la gráfica no coincidió con ninguna que había hecho anteriormente lo cual me genera bastante curiosidad.

Lunes, 29 Abril 2019

1. Papers de nuevos integradores symplecticos que thomas me paso

En los papers que thomas me mando usan unos integradores symplecticos de segundo orden llamados $SABA_2$ y $SBAB_2$, estos algoritmos pueden ser generalizados para cualquier orden en Hamiltonianos de la forma:

$$H = A + \epsilon B \quad (0.18)$$

Dadas las ecuaciones de Hamilton definidas por:

$$\frac{dp_n}{dt} = -\frac{\partial H}{\partial q_n}, \quad \frac{dq_n}{dt} = \frac{\partial H}{\partial p_n}, \quad n = 1, 2, \dots, N \quad (0.19)$$

Tambien recordamos los corchetes de poisson para funciones $f(p, q)$ y $g(p, q)$ los cuales son dados por:

$$\{f, g\} = \sum_{n=1}^n \left(\frac{\partial f}{\partial p_n} \frac{\partial g}{\partial q_n} - \frac{\partial f}{\partial q_n} \frac{\partial g}{\partial p_n} \right) \quad (0.20)$$

Entonces las ecuaciones de Hamilton toman la forma de:

$$\frac{dx}{dt} = \{H, x\} = L_H x \quad (0.21)$$

Donde L_H es el operador diferencial definido por $L_\chi f = \{\chi, f\}$. La solucion a la ecuacion anterior para condiciones iniciales $x(0) = x_0$ esta dada por:

$$x(t) = \sum_{n \geq 0} \frac{t^n}{n!} L_H^n x_0 = e^{tL_H} x_0 \quad (0.22)$$

Entonces los integradores quedaran:

$$SABA_2 = e^{c_1 \tau L_A} e^{d_1 \tau L_{\epsilon B}} e^{c_2 \tau L_A} e^{d_1 \tau L_{\epsilon B}} e^{c_1 \tau L_A} \quad (0.23)$$

con $c_1 = \frac{1}{2}(1 - \frac{1}{\sqrt{3}})$, $c_2 = \frac{1}{\sqrt{3}}$ y $d_2 = \frac{1}{2}$, mientras que el otro integrador queda:

$$SBAB_2 = e^{d_1 \tau L_{\epsilon B}} e^{c_2 \tau L_A} e^{d_2 \tau L_{\epsilon B}} e^{c_2 \tau L_A} e^{d_1 \tau L_{\epsilon B}} \quad (0.24)$$

con $c_2 = \frac{1}{2}$, $d_1 = \frac{1}{6}$ y $d_2 = \frac{2}{3}$. Al usar estos integradores estamos aproximando el comportamiento dinamico del Hamiltoniano real $H = A + \epsilon B$ por el Hamiltoniano $\tilde{H} = A + \epsilon B + \mathcal{O}(\tau^4 \epsilon + \tau^2 \epsilon^2)$.

La precision de los integradores puede ser mejorada al introducir el termino $C = \{\{A, B\}, B\}$ lo cual lleva a un sistema integrable. Esto agrega unos pequenos pasos que pueden ser agregados al integrador:

$$SABA_2C = e^{-(\tau^3\epsilon^2g/2)L_C}(SABA_2)e^{-(\tau^3\epsilon^2g/2)L_C} \quad (0.25)$$

Una expresion similar puede ser derivada para $SBAB_2$ donde el valor de g se escoge de tal manera que elimine la dependencia de $\tau^2\epsilon^2$ asi el integrador queda con el orden de $\mathcal{O}(\tau^4\epsilon + \tau^4\epsilon^2)$. En particular tenemos que $g = (2 - \sqrt{3})/24$ para $SABA_2$ y $g = 1/72$ para $SBAB_2$. Se enfatiza que estos integradores solo involucran pasos hacia adelante lo cual incrementa la estabilidad numerica.

Para nuestro sistema es necesario entonces calcular los corchetes de poisson de estos casos, esto tiene como resultado:

$$A = \frac{p^2}{2m} + \sum_{n=1}^N \frac{P_n^2}{2M_n} \quad (0.26)$$

$$B = -a\frac{x^2}{2} + b\frac{x^4}{4} + \sum_{n=1}^N M_n\omega_n^2 \frac{X_n^2}{2} + x \sum_{n=1}^N g_n X_n \quad (0.27)$$

$$L_A x = \frac{p}{m} \quad (0.28)$$

$$L_A X_n = \frac{P_n}{M_n} \quad (0.29)$$

$$L_A p = L_A P = 0 \quad (0.30)$$

$$L_B x = L_B X = 0 \quad (0.31)$$

$$L_B p = ax - bx^3 - \sum_{n=1}^N g_n X_n \quad (0.32)$$

$$L_B P_n = -M_n\omega_n^2 X_n - xg_n \quad (0.33)$$

Con esto, obtengo que:

$$\{\{A, B\}, B\} = \frac{(-ax + bx^3 + \sum_{n=1}^N g_n X_n)^2}{m} + \sum_{n=1}^N \frac{(xg_n + M_n\omega_n^2 X_n)^2}{M_n} \quad (0.34)$$

Por lo tanto:

$$\begin{aligned} L_C p &= \frac{-2}{m}(-ax + bx^3 + \sum_{n=1}^N g_n X_n)(-a + 3bx^2) - \sum_{n=1}^N \frac{2}{M_n}(xg_n + M_n\omega_n^2 X_n)(g_n) = \\ &= \frac{-2}{m}(a^2x - abx^3 - a \sum_{n=1}^N g_n X_n - 3abx^3 + 3b^2x^5 + 3bx^2 \sum_{n=1}^N g_n X_n) - \sum_{n=1}^N \frac{2}{M_n}(xg_n^2 + M_n g_n \omega_n^2 X_n) \end{aligned}$$

(0.35)

$$\begin{aligned} L_C P &= \frac{-2}{m}(-ax + bx^3 + \sum_{n'=1}^N g_{n'} X_{n'})(g_n) - 2(xg_n + M_n \omega_n^2 X_n) \omega_n^2 = \\ &= \frac{-2}{m}(-axg_n + bx^3g_n + \sum_{n'=1}^N g_{n'} X_{n'} g_n) - 2xg_n \omega_n^2 - 2M_n \omega_n^4 X_n \end{aligned} \quad (0.36)$$

Entonces tengo como resultado las evoluciones de la manera siguiente:

$$e^{\tau L_A} : \begin{cases} x' = \frac{p}{m} \tau + x \\ p' = p \\ X'_n = \frac{P_n}{M_n} \tau + X_n \\ P'_n = P_n \end{cases} \quad (0.37)$$

$$e^{\tau L_B} : \begin{cases} x' = x \\ p' = (ax - bx^3 - \sum_{n=1}^N g_n X_n) \tau + p \\ X'_n = X_n \\ P'_n = (-M_n \omega_n^2 X_n - xg_n) \tau + P_n \end{cases} \quad (0.38)$$

$$e^{-\tau L_C} : \begin{cases} x' = x \\ p' = \left(\frac{-2}{m} (a^2 x - abx^3 - a \sum_{n=1}^N g_n X_n - 3abx^3 + 3b^2 x^5 + 3bx^2 \sum_{n=1}^N g_n X_n) \right. \\ \quad \left. - \sum_{n=1}^N \frac{2}{M_n} (xg_n^2 + M_n g_n \omega_n^2 X_n) \right) \tau + p \\ X'_n = X_n \\ P'_n = \left(\frac{-2}{m} (-axg_n + bx^3g_n + \sum_{n'=1}^N g_{n'} g_n X_n) - (2xg_n \omega_n^2 + 2M_n \omega_n^4 X_n) \right) \tau + P_n \end{cases} \quad (0.39)$$

Lunes, 10 Marzo 2020

1. La era de Julia

Mud los integradores a Julia, en este lenguaje los pude generalizar facilmente y se me es mucho ms fcil agregar osciladores.

2. This is yet another example experiment

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Friday, 26 March 2010

1. This shows a sample table

Groups	Treatment X	Treatment Y
1	0.2	0.8
2	0.17	0.7
3	0.24	0.75
4	0.68	0.3

Cuadro 1: The effects of treatments X and Y on the four groups studied.

Table 1 shows that groups 1-3 reacted similarly to the two treatments but group 4 showed a reversed reaction.

Saturday, 27 March 2010

1. Bulleted list example

This is a bulleted list:

- Item 1
- Item 2
- ... and so on

2. This is an example experiment

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

3. This is another example experiment

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Formulae and Media Recipes

Media

Media 1

Compound	1L	0.5L
Compound 1	10g	5g
Compound 2	20g	10g

Cuadro 1: Ingredients in Media 1.

Formulae

Formula 1 - Pythagorean theorem

$$a^2 + b^2 = c^2 \text{ [diaconisDynamicalBiasCoin2007]}$$

bib.bib