Universidad de Sonora

Departamento de ciencias exactas

Física Computacional

Sistema de resortes acoplados

Alumno:
Martha Anahí Iñiguez Beltrán

214202804

20 de Marzo del 2018

Contents

1 Introducción

En esta breve actividad continuamos con el artículo de Temple H. Fay y Sarah Duncan Graham de las ecuaciones de dos resortes acoplados.

Continuaremos con las secciones 3 y 4 sobre un sistema de resortes no lineales y forzados.

La sección 3, supone que los resortes ya no cumplen la Ley de Hooke, sino que se comporta como un sistema no lineal. Los fenómenos no lineales abren toda una gran área de estudio en la Física. De hecho la mayoría de los fenómenos naturales son no lineales.

La sección 4, impone un forzamiento periódico sobre el sistema oscilador de masas.

2 Síntesis: Añadiendo la No Linearidad

Asumiendo que las fuerzas restauradoras son no lineales y tienen la forma $-kx+\mu x^3$, entonces las ecuaciones quedan:

$$m_1\ddot{x_1} = -\delta_1\dot{x}_1 - k_1x_1 + \mu_1x_1^3 - k_2(x_1 - x_2) + \mu_2(x_2 - x_1)^3$$
$$m_2\ddot{x_2} = -\delta_2\dot{x}_2 - k_2(x_2 - x_1) + \mu_2(x_2 - x_1)^3$$

La solución de este tipo de sistemas es mucho más complicada que la de los sistemas lineales. A continuación se muestran algunos ejemplos.

A continuación agregamos el código utilizado para mostrar éste ejemplo:

```
def vectorfield(w, t, p):

    x1, y1, x2, y2 = w
    m1, m2, k1, k2, b1, b2, n1, n2 = p

    # Create f = (x1',y1',x2',y2'):
    f = [y1,(-b1 * y1 - k1 * x1 + n1 * (x1 ** 3) - k2 * (x1-x2) + n2 * (x1 - x2) **
    return f
```

2.1 Ejemplo 3.1

Asuma que $m_1 = m_2 = 1$.Describe el movimiento para resortes con constantes $k_1 = 0.4$ y $k_2 = 1.808$, coeficientes de amortiguamiento $\delta_1 = 0$ y $\delta_2 = 0$, coeficientes de no linealidad $\mu_1 = -1/6$ y $\mu_2 = -1/10$, y con condiciones iniciales $(x_1(0), \dot{x_1}(0), x_2(0), \dot{x_2}(0))$=(1,0,-1/2,0).

Cómo no hay amortiguamiento, el movimiento es oscilatorio y aparenta ser periódico. Por la no linealidad, el movimiento es muy sensible a condiciones iniciales.

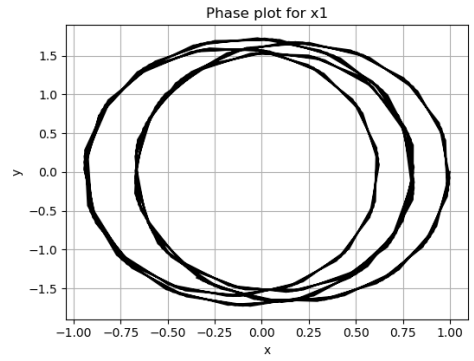
```
# Use ODEINT to solve the differential equations defined by the vector field
from scipy.integrate import odeint

# Parameter values
# Masses:
m1 = 1.0
m2 = 1.0

# Spring constants
k1 = 0.4
k2 = 1.808

# Friction coefficients
b1 = 0.0
b2 = 0.0


#Coeficientes no lineales
n1=-1/6
n2=-1/10

# Initial conditions
# x1 and x2 are the initial displacements; y1 and y2 are the initial velocities
x1 = 1.0
y1 = 0.0
x2 = -1/2
y2 = 0.0

# ODE solver parameters
abserr = 1.0e-8
```

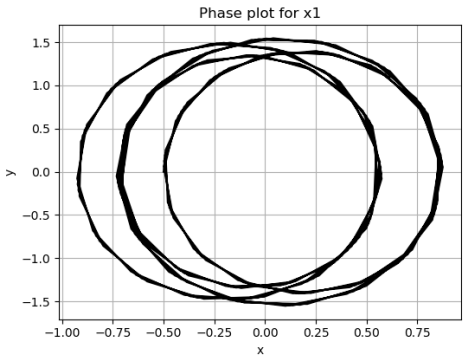
```
relerr = 1.0e-6
stoptime = 50.0
numpoints = 250

# Create the time samples for the output of the ODE solver.
# I use a large number of points, only because I want to make
# a plot of the solution that looks nice.
t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

# Pack up the parameters and initial conditions:
p = [m1, m2, k1, k2, b1, b2, n1, n2]
w0 = [x1, y1, x2, y2]

# Call the ODE solver.
wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('nonlinear3.1.dat', 'w') as f:
    # Print & save the solution.
    for t1, w1 in zip(t, wsol):
        print (t1, w1[0], w1[1], w1[2], w1[3], file=f)
```

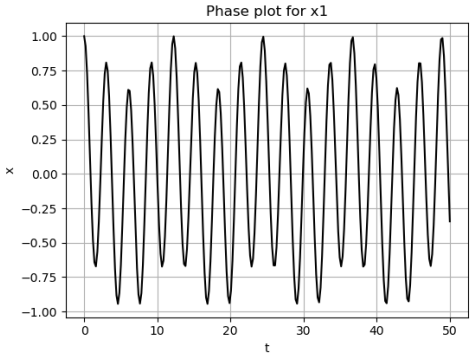
2.2 Ejemplo 3.2

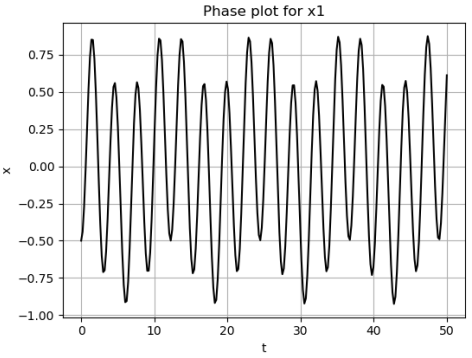
Asuma que $m_1 = m_2 = 1$.*Describe el movimiento para resortes con constantes* $k_1 = 0.4$ *y* $k_2 = 1.808$, *coeficientes de amortiguamiento* $\delta_1 = 0$ *y* $\delta_2 = 0$, *coeficientes de no linealidad* $\mu_1 = -1/6$ *y* $\mu_2 = -1/10$, *y con condiciones iniciales* $(x_1(0), \dot{x_1}(0), x_2(0), \dot{x_2}(0))$*=(-0.5,1/2,3.001,5.9).*

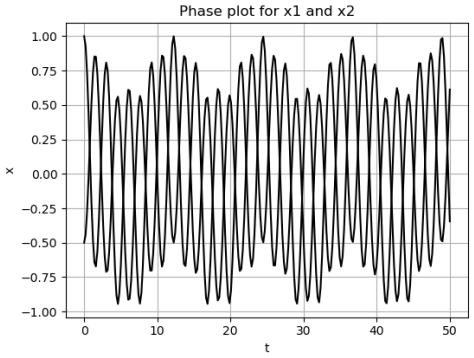
```
def vectorfield(w, t, p):

    x1, y1, x2, y2 = w
    m1, m2, k1, k2, L1, L2, b1, b2, n1, n2 = p

    f = [y1,
         (-b1 * y1 - k1 * (x1 - L1) + n1 * (x1 - L1)**3 + k2 * (x2 - x1 - L2) + n2
         y2,
         (-b2 * y2 - k2 * (x2 - x1 - L2) + n2 * (x2 - x1)**3) / m2]
    return f
```

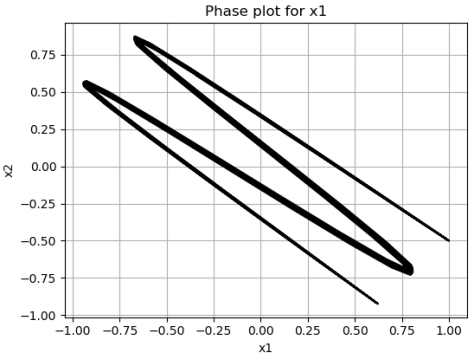

Phase plot for x1
y
x


Phase plot for x1
y
x

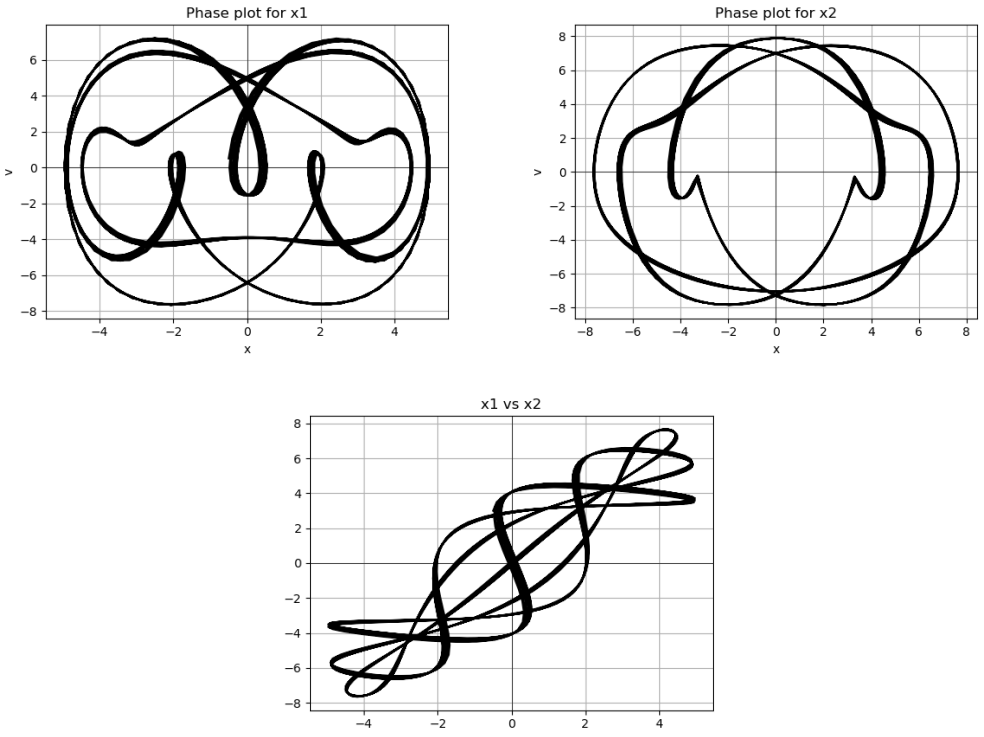

Phase plot for x1
x
t


Phase plot for x1
x
t


Phase plot for x1 and x2
x
t


Phase plot for x1
x2
x1

```
#Use ODEINT function
from scipy.integrate import odeint

m1 = 1.0
m2 = 1.0

k1 = 0.4
k2 = 1.808

L1 = 0.0
L2 = 0.0

b1 = 0.0
b2 = 0.0

n1 = -(1.0/6.0)
n2 = -(1.0/10.0)

x1 = -0.5
y1 = 0.5
x2 = 3.001
y2 = 5.9

abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 200.0
numpoints = 2150


t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

p = [m1, m2, k1, k2, L1, L2, b1, b2, n1, n2]
w0 = [x1, y1, x2, y2]

wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('nonlinear3.2.dat', 'w') as f:
    for t1, w1 in zip(t, wsol):
```




```
print (t1, w1[0], w1[1], w1[2], w1[3],file=f)
```

2.3 Ejemplo 3.3

Asuma que $m_1 = m_2 = 1$.*Describe el movimiento para resortes con constantes* $k_1 = 0.4$ *y* $k_2 = 1.808$, *coeficientes de amortiguamiento* $\delta_1 = 0$ *y* $\delta_2 = 0$, *coeficientes de no linealidad* $\mu_1 = -1/6$ *y* $\mu_2 = -1/10$, *y con condiciones iniciales* $(x_1(0), \dot{x_1}(0), x_2(0), \dot{x_2}(0))$*=(-0.6,1/2,3.001,5.9).*

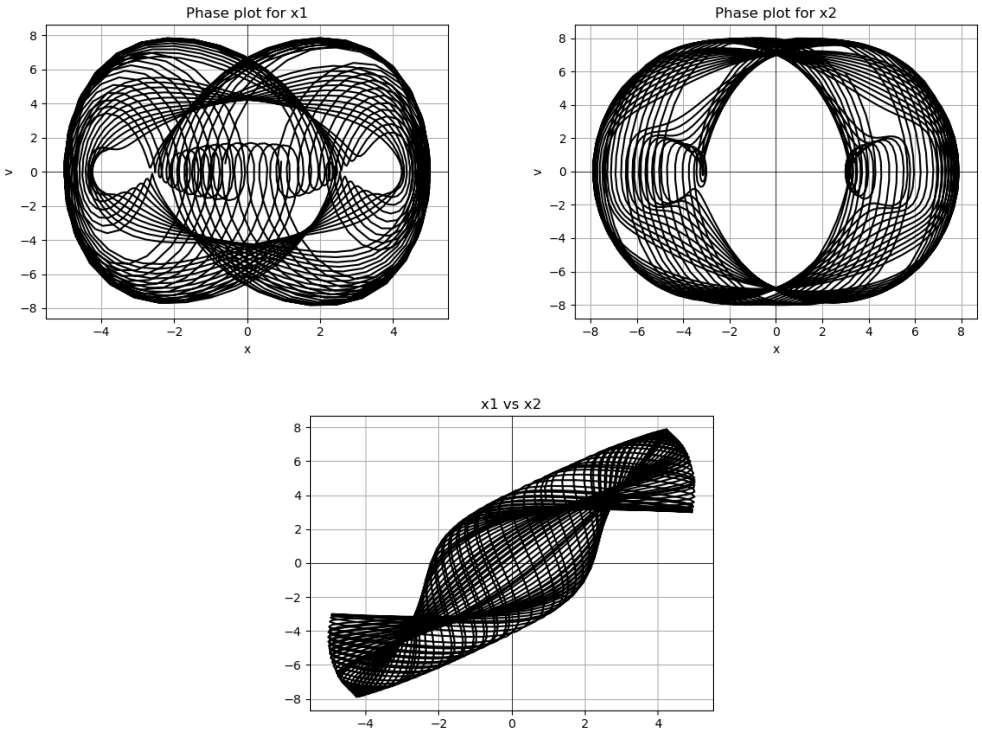
```
from scipy.integrate import odeint


m1 = 1.0
m2 = 1.0

k1 = 0.4
k2 = 1.808
```

```
L1 = 0.0
L2 = 0.0

b1 = 0.0
b2 = 0.0

n1 = -(1.0/6.0)
n2 = -(1.0/10.0)

x1 = -0.6
y1 = 0.5
x2 = 3.001
y2 = 5.9

abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 200.0
numpoints = 2150

t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

p = [m1, m2, k1, k2, L1, L2, b1, b2, n1, n2]
w0 = [x1, y1, x2, y2]

wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('nonlinear3.3.dat', 'w') as f:
    for t1, w1 in zip(t, wsol):
        print (t1, w1[0], w1[1], w1[2], w1[3],file=f)
```

3 Añadiento Forzamiento

Se puede añadir una fuerza externa de una manera muy sencilla al modelo, siendo o no igual para ambas masas. Suponiendo que se aplica una fuerza senoidal con forma $F\cos(wt)$, las ecuaciones quedan:

$m_1\ddot{x_1} = -\delta_1\dot{x}_1 - k_1x_1 + \mu_1x_1^3 - k_2(x_1 - x_2) + \mu_2(x_2 - x_1)^3 + F_1\cos(w_1t)$



$m_2\ddot{x_2} = -\delta_2\dot{x}_2 - k_2(x_2 - x_1) + \mu_2(x_2 - x_1)^3 + F_2\cos(w_2 t)$

El rango de movimientos para un sistema no lineal es bastante amplio. Las condiciones para que este movimiento ocurra son muy difíciles de especificar, por lo que se muestra un ejemplo:

3.1 Ejemplo 4.1

Asuma que $m_1 = m_2 = 1$.*Describe el movimiento para resortes con constantes* $k_1 = 2/5$ *y* $k_2 = 1$, *coeficientes de amortiguamiento* $\delta_1 = 1/10$ *y* $\delta_2 = 1/5$, *coeficientes de no linealidad* $\mu_1 = 1/6$ *y* $\mu_2 = 1/10$, *fuerzas de amplitud* $F_1 = 1/3$ *y* $F_2 = 1/5$, *frecuencias de forzamiento* $w_1 = 1$ *y* $w_2 = 3/5$ *y con condiciones iniciales* $(x_1(0), \dot{x_1}(0), x_2(0), \dot{x_2}(0))$*=(0.7,0,0.1,0).*

```
import numpy as np
def vectorfield(w, t, p):

    x1, y1, x2, y2 = w
    m1, m2, k1, k2, L1, L2, b1, b2, n1, n2, F1, F2, w1, w2 = p
```

```
    #Creamos f = (x1',y1',x2',y2')
    f = [y1,
         (-b1 * y1 - k1 * (x1 - L1) + n1 * (x1 - L1)**3 + k2 * (x2 - x1 - L2) + n2
         y2,
         (-b2 * y2 - k2 * (x2 - x1 - L2) + n2 * (x2 - x1)**3 + F2 * np.cos(w2*t)) /
    return f

from scipy.integrate import odeint

m1 = 1.0
m2 = 1.0

k1 = (2.0/5.0)
k2 = 1

L1 = 0.0
L2 = 0.0

b1 = (1.0/10.0)
b2 = (1.0/5.0)

n1 = (1.0/6.0)
n2 = (1.0/10.0)

F1 = (1.0/3.0)
F2 = (1.0/5.0)

w1 = 1
w2 = (3.0/5.0)

# Condiciones iniciales
x1 = 0.7
y1 = 0
x2 = 0.1
y2 = 0

abserr = 1.0e-8
relerr = 1.0e-6
stoptime = 250.0
```

```
numpoints = 2150

t = [stoptime * float(i) / (numpoints - 1) for i in range(numpoints)]

p = [m1, m2, k1, k2, L1, L2, b1, b2, n1, n2, F1, F2, w1, w2]
w0 = [x1, y1, x2, y2]

wsol = odeint(vectorfield, w0, t, args=(p,),
              atol=abserr, rtol=relerr)

with open('addedforce4.1.dat', 'w') as f:
    for t1, w1 in zip(t, wsol):
        print (t1, w1[0], w1[1], w1[2], w1[3],file=f)
```

4 Conclusión

Concluyendo la actividad pudimos obtener las gráficas totalmente similares a las del artículo. El uso de python y el ambiente de jupyter lab facilitó el desarrollo del código y aquí es donde finalizamos con los ejemplos del artículo.

5 Bibliografía

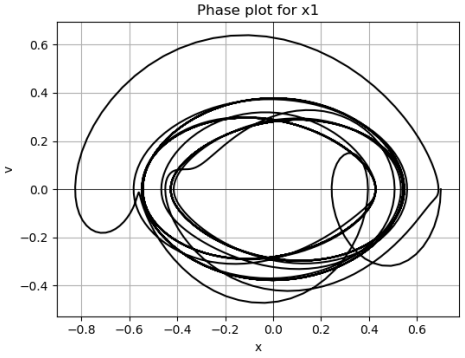
- H. FAY, T., & DUNCAN GRAHAM, S. (2003). Coupled Spring ecuations (pp. 65-79). Int. J. Math. Educ. Sci. Technol. Recuperado el 20 de Marzo del 2018 desde http://math.oregonstate.edu/ gibsonn/Teaching/MTH323-010S15/Supplements/coupled_spring.pdf

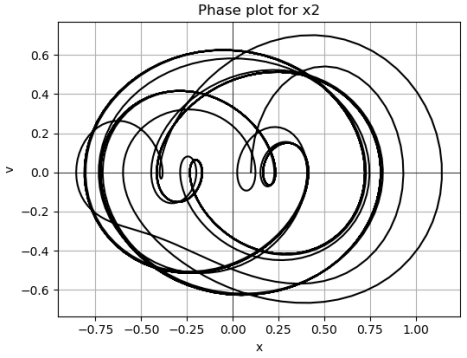
6 Apéndice

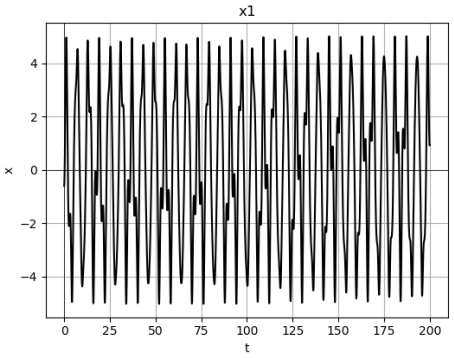
¿Qué más te llama la atención de la actividad completa? ¿Que se te hizo menos interesante?

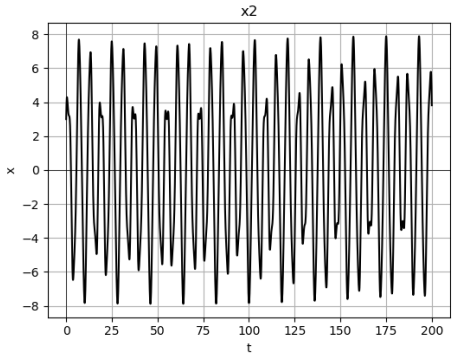
Me gustó toda la actividad. Fue muy interesante.

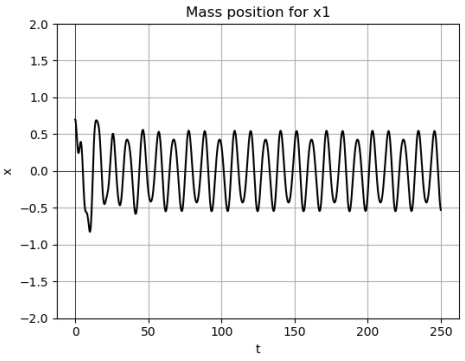
¿De un sistema de masas acopladas como se trabaja en esta actividad, hubieras pensado que abre toda una nueva área de fenómenos no lineales?

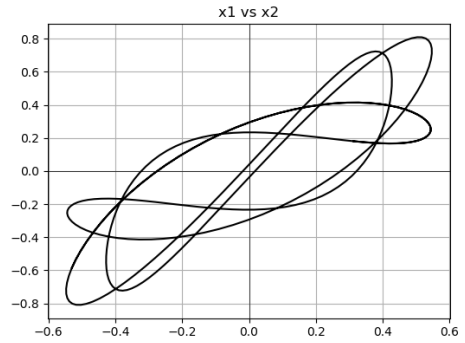

Phase plot for x1
v
x

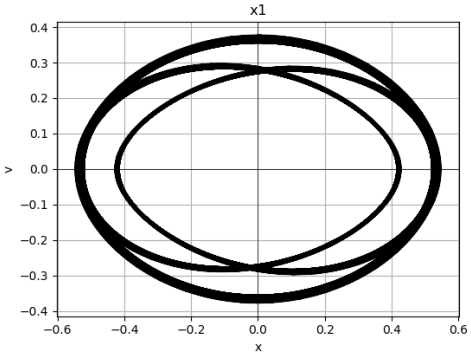

Phase plot for x2
v
x

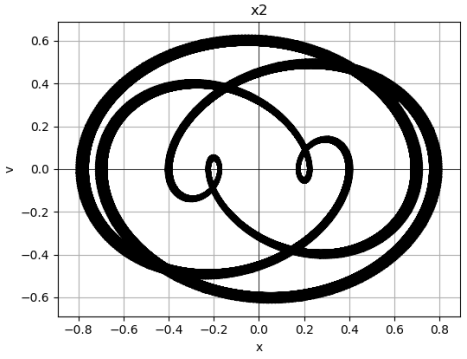

x1
x
t


x2
x
t


Mass position for x1
x
t


x1 vs x2


x1
v
x


x2
v
x

Si, puesto que en la naturaleza aparece mas fenómenos no lineales.

¿Qué propondrías para mejorar esta actividad? ¿Te ha parecido interesante este reto?

En mi opinion la actividad está completa y es interesante.

¿Quisieras estudiar mas este tipo de fenómenos no lineales?

Si sería interesante.