Métodos Computacionales

Tarea 3 - Ecuaciones Diferenciales Ordinarias

12-09-2016

La solución a este taller debe subirse por SICUA antes de las 8:00AM del jueves 6 de octubre del 2016.
(10 pt) Los códigos deben subirse en un único archivo `.zip` con el nombre `NombreApellido_hw3.zip`. Por ejemplo yo debería subir el zip `JaimeForero_hw3.zip`. Al descomprimir el zip debe crearse la carpeta `JaimeForero_hw3` y adentro deben estar los códigos solicitados. Ningún programa puede utilizar las funciones especiales de numpy, scipy o scikit-learn para integrar, diferenciar, encontrar ceros, resolver sistemas de ecuaciones o hacer PCA.

1. **Osciladores acoplados**

   Considere un sistema de N masas atadas por una cuerda como se muestran en la figura 1.

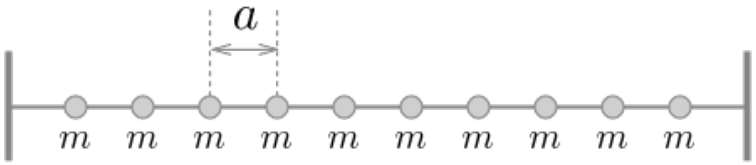

Figura 1: Cuerda cargada.

Para separaciones pequeñas alrededor del punto de equilibrio, la tensión T a lo largo de la cuerda se puede tomar como constante. Según la figura 2, las ecuaciones de movimiento están dadas por

$$\frac{\mathrm{d}y'_n}{\mathrm{d}t} = -\frac{T}{ma}\left(2y_n(t) - y_{n-1}(t) - y_{n+1}(t)\right),$$

$$y'_n = \frac{\mathrm{d}y_n}{\mathrm{d}t},$$

para n entero tal que $0 \leq n \leq N$, con condiciones de frontera fijas $y_0(t) = y_{N+1}(t) = 0$.

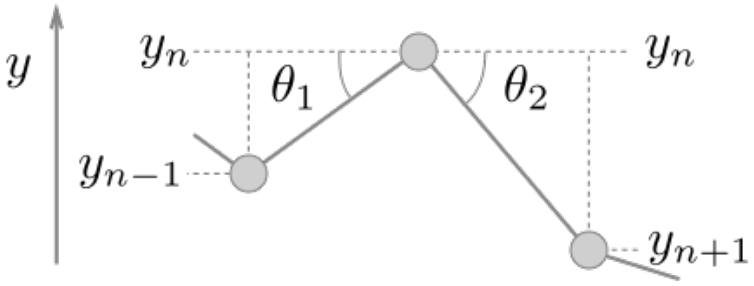

Figura 2: Sección de la cuerda cargada.

Escriba un programa en python (`acoplado.py`) que lea un archivo de texto en el siguiente formato

```
N
T
m
a
y10 y20 y30 ... yN0
y1p0 y2p0 y3p0 ... yNp0
```

Donde N es el número de osciladores, T, m y a los parámetros ya mencionados y las últimas dos líneas representan las condiciones iniciales para las posiciones y las velocidades de cada una de las masas (separadas por espacios).

El programa debe resolver el sistema de ecuaciones para un intervalo de tiempo que permita visualizar el comportamiento (de tal manera que no sea menor a un período del movimiento ni innecesariamente largo) y manteniendo las pérdidas de energía por debajo del 10 % de la energía total inicial. Al final el programa debe generar:

- (20 puntos) Un gif animado de nombre `movimiento.gif` la evolución temporal de y_n contra n para todas las masas en función del tiempo.
- (10 puntos) Una gráfica llamada `energia.pdf` de la variación de la energía total del sistema respecto a la energía inicial, $(E(t) - E_0)/E_0$, como función del tiempo.

El código debe poder ejecutarse como

```
python acoplado.py condiciones.dat
```

donde `condiciones.dat` es el archivo que describe las condiciones iniciales.

2. **Ciclos**
La siguiente dirección `https://archive.ics.uci.edu/ml/machine-learning-databases/00360/` contiene datos correspondientes a mediciones de calidad del aire. Los datos fueron tomados por Saverio De Vito de la *National Agency for New Technologies, Energy and Sustainable Economic Development* en Italia. El archivo contiene 9358 mediciones de promedios hechos cada hora de mediciones de una serie de cinco sensores químicos diferentes. Los sensores estaban localizados a nivel de suelo en una calle contaminada de una ciudad italiana. Los datos fueron grabados entre Marzo 2004 y Febrero 2005. Una descripción detallada de los datos se encuentra en `https://archive.ics.uci.edu/ml/datasets/Air+Quality`.

(30 puntos) Escriba un programa en python (`ciclos.py`) que lea el archivo `AirQualityUCI.csv` y haga un análisis de Fourier para obtener dos períodos dominantes (en unidades de días) que se encuentren en la variabilidad de los datos de las mediciones reales de las concentraciones de

- CO.
- Hydrocarburos diferentes a Metano.
- Benceno.
- NOx.
- NO2.

Los resultados deben escribirse en un archivo `periodos.dat` que contiene dos columnas (para cada uno de los períodos) y cinco filas (para cada uno de las concentraciones).

El código debe poder ejecutarse como

```
python ciclos.py AirQualityUCI.csv
```

3. **Mezclas**

Utilizando el mismo conjunto de datos del punto anterior escriba un código en python (`mezclas.py`) que haga un análisis PCA y encuentre las dos primeras componentes principales a partir de las mismas columnas anteriores.

El programa debe:

- (10 puntos) Hacer una gráfica `pca.pdf` donde se muestren todas las mediciones en el plano correspondiente a las dos variables PCA principales.
- (10 puntos) Escribir en un archivo `pca.dat` las dos columnas correspondientes a las componentes principales.
- (10 puntos) Escribir en un archivo `varianza.dat` el porcentaje de la varianza que explican las dos primeras componentes principales.

Cuidado. Las diferentes concentraciones no se miden en las mismas unidades. ¿Cómo hacer para que las mediciones sean entonces comparables y las componentes PCA no dependan de las unidades utilizadas en la medición?

El código debe poder ejecutarse como

```
python mezclas.py AirQualityUCI.csv
```