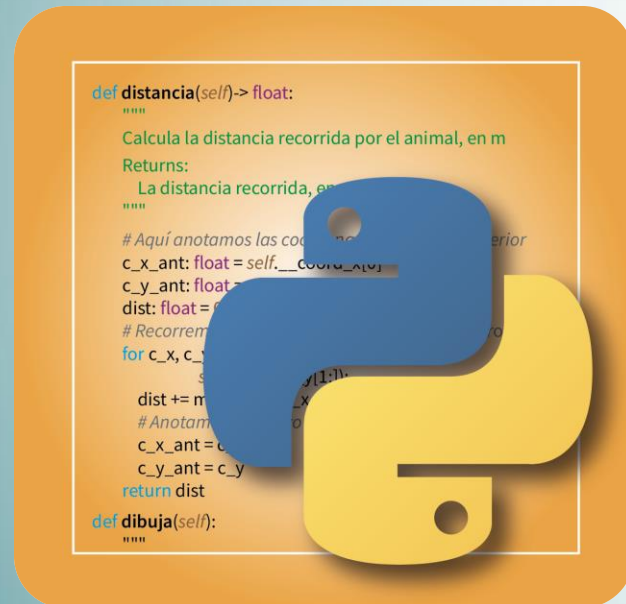


Programación

Práctica 12. Oscilador amortiguado



Michael González Harbour
José Javier Gutiérrez García
José Carlos Palencia Gutiérrez
José Ignacio Espeso Martínez
Adolfo Garandal Martín

Departamento de Ingeniería
Informática y Electrónica

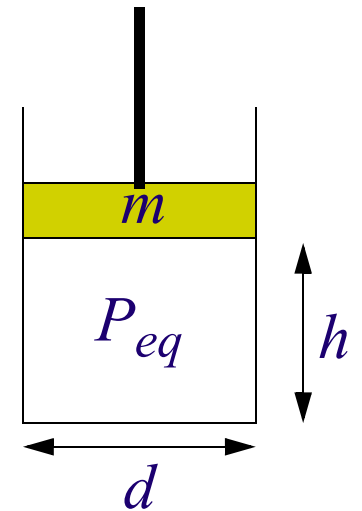
Este material se publica con licencia:
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 12: Oscilador amortiguado

Objetivos: Practicar con `numpy`, lectura de ficheros de texto y excepciones

También queremos practicar el ajuste de los parámetros de una función arbitraria a partir de una colección de datos experimentales

Descripción: El método de *Rüchardt* es un procedimiento que permite obtener la constante adiabática de un gas mediante la medida de la frecuencia de las oscilaciones de un émbolo cilíndrico de masa m que cierra un recipiente de altura h y diámetro d que contiene ese gas, cuando se saca el émbolo de su posición de equilibrio a presión P_{eq}



Una forma de obtener estas oscilaciones es medir la presión interna del recipiente

Ecuación de la sobrepresión

Es una oscilación armónica que en un experimento real es amortiguada, por lo que su forma funcional será:

$$\Delta P = A \cdot e^{-\beta t/2} \cdot \cos(\omega t + \delta) + \Delta P_{emb}$$

- ΔP es la sobrepresión del gas (en kPa) en relación a la presión atmosférica
- A es la amplitud de las oscilaciones (en kPa)
- β es el coeficiente de amortiguamiento (en 1/s)
- ω es la frecuencia angular de las oscilaciones (en rad/s)
- δ es el desfase (en rad)
- ΔP_{emb} es la sobrepresión producida por el peso del émbolo (en kPa)
- t es el tiempo (en s)

Ecuación del coeficiente adiabático

Una vez obtenidas la frecuencia angular ω , el coeficiente de amortiguamiento β y la sobrepresión ΔP_{emb} , podemos obtener el coeficiente adiabático, γ (sin unidades):

$$\gamma = \frac{mh(\omega^2 + \beta^2/4)}{\pi(d/2)^2 P_{eq}}$$

- m : masa del émbolo
- h : altura de la columna de gas
- d : diámetro del émbolo
- P_{eq} : Presión de equilibrio = presión atmosférica + ΔP_{emb}

Para hacer el cálculo utilizar unidades compatibles, por ejemplo del sistema internacional

Módulo `ruchardt.py`: función

Se pide escribir el módulo `ruchardt.py` que contenga una implementación de la ecuación de la sobrepresión y un programa `main` que ajuste datos experimentales a la misma

Se pretende obtener el mejor ajuste para los parámetros A , β , ω , δ y ΔP_{emb}

Función de la sobrepresión: El primer parámetro debe ser la variable independiente (es decir, el tiempo) y el resto serán los parámetros que queremos ajustar

- La función retornará el valor ΔP de la ecuación de la sobrepresión
- Como la función que va a realizar el ajuste trabajará con arrays de `numpy`, las funciones matemáticas que usemos deben ser de `numpy`, y no de `math`

Módulo `ruchardt.py`: `main`

El programa principal hará las siguientes operaciones:

- Leerá los datos experimentales, que se proporcionan en el fichero de texto `butano.txt`, y que tiene
 - una línea de encabezamiento con los títulos de las columnas, que se deberá ignorar
 - sucesivas líneas con parejas de valores numéricos separados por tabuladores (carácter `\t`): tiempo, en *s* y sobrepresión, en *kPa*
 - se puede suponer que el fichero es correcto y no tiene errores
- Se deberá tratar la excepción que se lanza si el fichero no existe, abandonando el resto de los pasos del `main` y mostrando un mensaje de error en pantalla
- Los datos numéricos leídos se guardarán en sendos arrays de `numpy`
- No olvidar que es preciso cerrar el fichero tras su lectura

Módulo `ruchardt.py`: `main` (cont.)

- El `main` realizará el ajuste, utilizando para ello la función `curve_fit` que está en `scipy.optimize`. Permite hacer un ajuste de una función arbitraria usando mínimos cuadrados no lineales
- A esta función hay que darle como parámetros el nombre de la función a ajustar, los dos arrays con los datos de las variables independiente (tiempo) y dependiente (sobrepresión), y una tupla con valores iniciales de los parámetros cercanos a la solución buscada
 - p.e., $A=6\text{kPa}$, $\beta=10\text{s}^{-1}$, $\omega=200\text{rad/s}$, $\delta=-1\text{rad}$ y $\Delta P_{emb}=0.2\text{kPa}$
- la función retorna una tupla con:
 - Array de `numpy` con los valores óptimos de los parámetros a ajustar (A , β , ω , δ y ΔP_{emb} en este caso)
 - Array bidimensional de `numpy` conteniendo la covarianza estimada para el ajuste. La diagonal proporciona la varianza en la estimación de cada parámetro

Módulo `ruchardt.py`: `main` (cont.)

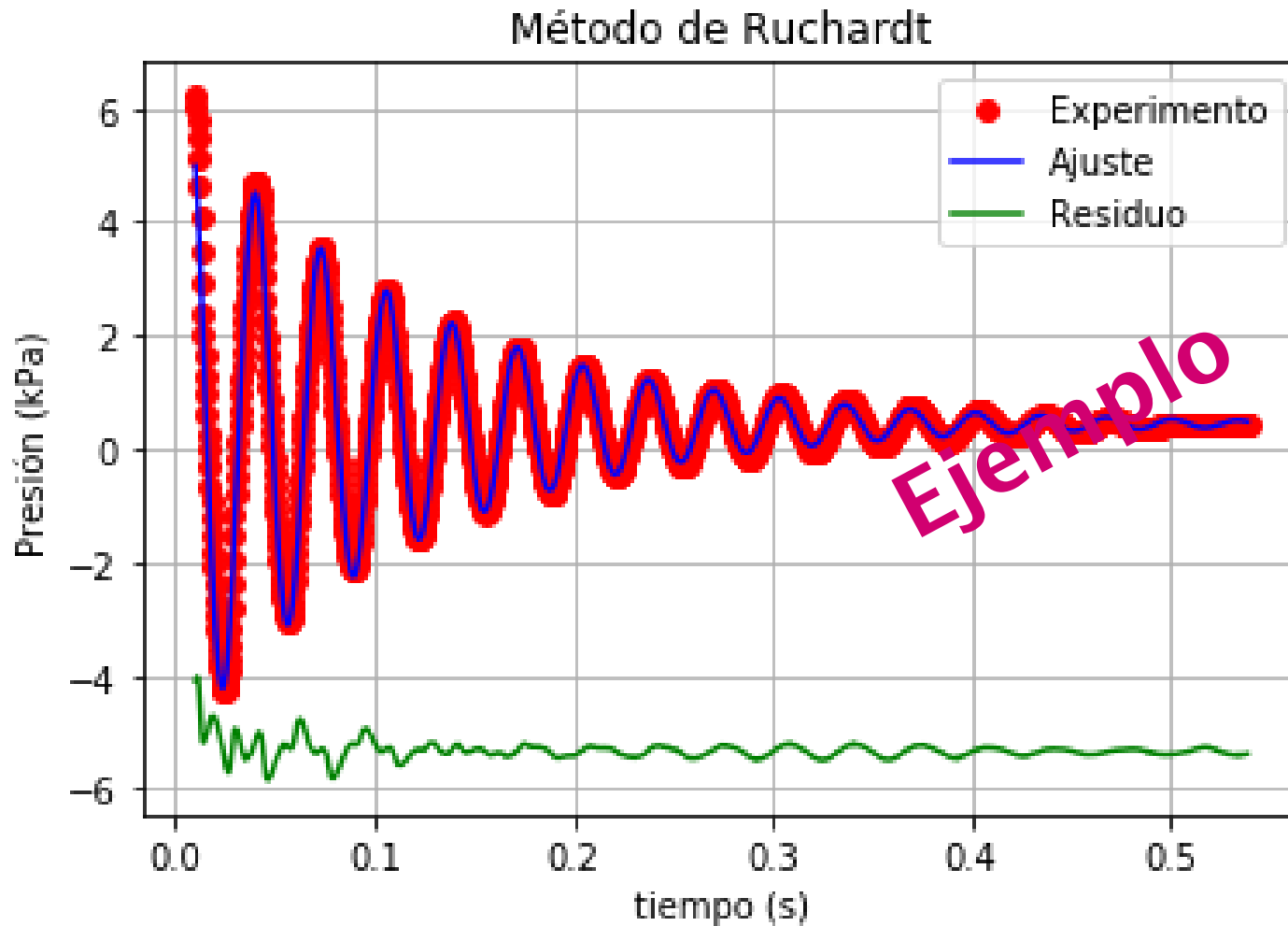
- El `main` representará gráficamente (`pyplot`) los datos experimentales junto con el ajuste
 - el ajuste consiste en obtener para cada tiempo de las medidas experimentales el valor devuelto por la función de la sobrepresión cuando se usan los parámetros obtenidos en el ajuste
 - esto puede hacerse sin bucles, operando directamente con los arrays de `numpy`
- Luego mostrará los valores de los parámetros obtenidos en el ajuste
- Finalmente pedirá por teclado los valores de m , d y h y mostrará en pantalla el valor del coeficiente adiabático γ
 - para la presión de equilibrio = presión atmosférica + ΔP_{emb} se usará una presión atmosférica de 101.3 kPa

Parte avanzada

Añadir al software desarrollado el análisis de errores y residuos

- los errores estándar son las raíces cuadradas de los elementos de la diagonal de la matriz de covarianza
 - se mostrarán estos errores junto a los valores de cada parámetro ajustado, usando el formato: valor \pm error
- los residuos son las diferencias entre cada presión experimental y la presión obtenida con la función de la presión ajustada
 - se hará una gráfica de los residuos en la misma gráfica donde se muestran los valores experimentales y el ajuste
 - con objeto de mostrar los residuos debajo de las otras gráficas se hará un desplazamiento de los valores de cada residuo sumándoles el siguiente término:
$$\text{valor_min} - 0.1 * (\text{valor_max} - \text{valor_min})$$
 - siendo **valor_min** y **valor_max** el mínimo y máximo de las presiones experimentales

Ejemplo de la gráfica obtenida tras el ajuste



Entregar 2 archivos

1. Código del módulo `ruchardt.py`
2. Informe en pdf con:
 - Captura de pantalla de la gráfica con los datos experimentales y el ajuste
 - Captura de pantalla de los resultados de la ejecución del `main` con los valores de los parámetros ajustados y el coeficiente adiabático para $m=35g$, $d=32.5mm$, $h=73mm$
 - Captura de pantalla de los resultados con excepciones. Cambiar el nombre al fichero y mostrar los resultados de la ejecución del `main` cuando salta la excepción
 - Si se ha hecho la parte avanzada las dos primeras capturas se mostrarán con los errores y residuos