

# Métodos Computacionales

## Tarea 4 — 2018–20

Los archivos:

`hw4.mk`,

`ODE.cpp`.

`PDE.cpp`,

`Plots_hw4.py`

y

`Resultados_hw4.tex`.

deben estar comprimidos en `ApellidoNombre_hw4.zip` y deben descomprimirse en un directorio `ApellidoNombre_hw4`. Este directorio comprimido debe contener únicamente los archivos mencionados anteriormente y debe subirlo a SICUA antes de las 8:00 pm del lunes 19 de noviembre de 2018 (5 puntos). Recuerde que es un trabajo completamente individual (no puede usar códigos que usted no haya escrito en su totalidad ni puede recibir ayuda de nadie cuando esté escribiendo su código). Recuerde también que los códigos deben correr sin botar errores en los computadores de compufísica (Y110B). Si su código no corre la nota de ese ejercicio será de cero. Para evitarlo comente la sección de código problemática. Recuerde que es buena práctica comentar el código y elegir buenos nombres para sus variables. Eso será tenido en cuenta durante el proceso de corrección de esta tarea. No se aceptarán trabajos entregados tarde ni por otro medio distinto a SICUA.

### 1. (5 points) **Github**

En esta tarea se evaluará nuevamente el uso de Github de la siguiente manera:

- Los archivos de solución deben, además de subirse a SICUA, estar en un repositorio cuyo enlace debe ser enviado por SICUA cuando se envíe la solución de la tarea.
- Debe haber al menos 15 commits de la tarea en dónde se vea claramente la **evolución** del trabajo (no es válido hacer varios "commits" justo antes de la hora de entrega de la tarea).
- Los commits se deben hacer para los archivos individuales de la tarea a medida que su trabajo en cada archivo vaya avanzando.

### 2. (30 points) **ODE**

En este ejercicio deben solucionar la ecuación de movimiento de un proyectil para el caso en que hay una fricción proporcional a  $v^2$ . Deben escribir un código en c++ llamado `ODE.cpp` donde implementen el método de Leap-Frog o el método de Runge Kutta de 4to orden para resolver la ecuación de movimiento:

$$\frac{d^2 \vec{x}(t)}{dt^2} = -\vec{g} - c \frac{|\vec{v}(t)|^2}{m} \frac{\vec{v}(t)}{|\vec{v}(t)|} \quad (1)$$

donde  $g = 10m/s^2$  es la aceleración gravitacional,  $c = 0.2$  es el coeficiente de fricción y  $m = 0.2\text{kg}$  es la masa del proyectil.

En la primera parte, tome como condición inicial  $\vec{x}(t=0) = (0,0)$  y una velocidad de magnitud 300 m/s y con un ángulo de 45 grados respecto a la horizontal.

Su código debe imprimir un mensaje que indique la distancia recorrida por el proyectil. Además su código debe guardar en un archivo las posiciones y velocidades del proyectil en función del tiempo (para después graficar la trayectoria del proyectil).

En la segunda parte pruebe ángulos de 10, 20, 30, 40, 50, 60, 70 grados. Su código debe imprimir un mensaje que indique para cuál de los ángulos anteriores la distancia recorrida por el proyectil es mayor. Además su código debe guardar en un segundo archivo las posiciones y velocidades del proyectil en función del tiempo para todos los ángulos (para después graficar dichas trayectorias del proyectil).

3. (30 points) **PDE.**

En este ejercicio deben solucionar la ecuación de difusión en 2 dimensiones.

$$\frac{\partial T(x, y)}{\partial t} = \nu \frac{\partial^2 T(x, y)}{\partial x^2} + \nu \frac{\partial^2 T(x, y)}{\partial y^2} \quad (2)$$

Suponga que tiene una varilla metálica de 10 cm de diámetro que está entre una roca. Suponga que la roca es una calcita con coeficiente de difusión  $\nu = k/(C_p \cdot \rho)$  donde  $k = 1.62 m^2 s^{-1}$  es la conductividad térmica,  $C_p = 820 J kg^{-1} K^{-1}$  es el calor específico y  $\rho = 2.71 g cm^{-3}$  es la densidad.

Suponga también que la temperatura de la varilla se mantiene siempre a 100 grados centígrados. Calcule la evolución de la temperatura en la calcita. Sólo considere una sección de roca 2D cuadrada de 50cm por 50cm alrededor de la varilla y perpendicular a ella.

En el caso 1 considere que tiene condiciones de frontera fijas a 10 grados centígrados.

En el caso 2 considere condiciones de frontera abiertas.

En el caso 3 considere condiciones de frontera periódicas.

Para todos los casos anteriores, calcule la evolución temporal de la temperatura de la calcita, evolucionando el sistema hasta que esre haya llegado a una configuración de equilibrio. Genere los datos necesarios para hacer una gráfica de las condiciones iniciales, dos gráficas de estados intermedios del sistema y una gráfica de la configuración de equilibrio.

Genere también los datos que le permitan hacer una gráfica de la temperatura **promedio** de la clacita en función del tiempo para los tres casos de condiciones de frontera.

4. (30 points) **Gráficas, makefiles y pdf de resultados.** El código `Plots_hw4.py` debe (8 puntos):

- Leer y guardar los datos generados por `PDE.cpp` y `ODE.cpp`
- ODEs: graficar la trayectoria del proyectil para 45 grados y la trayectoria del proyectil para los distintos angulos. Guarde dichas gráficas sin mostrarlas e inclúyalas en el pdf de resultados.

- PDEs: Para cada uno de los casos hacer una gráfica 3D de las condiciones iniciales, dos gráficas de estados intermedios del sistema y una gráfica de la configuración de equilibrio.

Haga también una gráfica de la temperatura **promedio** de la clacita en función del tiempo para los tres casos de condiciones de frontera.

- Bono (5 pts) Hacer una animación 3D de la temperatura del sistema en función del tiempo).

El archivo `Resultados_hw4.tex` debe (5 puntos):

- Organizar las gráficas obtenidas (**debe tener dos gráficas del ejercicio del proyectil y todas las gráficas del ejercicio de difusión térmica.**). Haga una sección por cada ejercicio de la tarea y describa brevemente sus resultados. En el ejercicio de difusión incluya una pequeña descripción del caso, las condiciones de frontera, explicación de las diferencias observadas para distintas condiciones de frontera, etc... Este archivo debe estar incluido dentro de las dependencias del makefile y debe permitir generar un archivo `Resultados_hw4.pdf`

El archivo `hw4.mk` debe (15 puntos):

- Incluir todas las dependencias y reglas necesarias para generar y actualizar el archivo: `Resultados_hw4.pdf`.  
Los archivos que deben subir a Sicua (comprimidos en `ApellidoNombre_hw4.zip`) son:  
`hw4.mk`,  
`ODE.cpp`.  
`PDE.cpp`,  
`Plots_hw4.py`  
y  
`Resultados_hw4.tex`.