ANÁLISIS NUMÉRICO I / ANÁLISIS NUMÉRICO – 2024
Trabajo de Laboratorio N$^{\underline{\text{o}}}$ 5

1. Programar una función en `python` que integre numéricamente usando las reglas compuestas del trapecio, punto medio y Simpson, nombrarla `intenumcomp`. La función deberá ejecutarse:

   ```
   python> S = intenumcomp(fun,a,b,N,regla)
   ```

   donde `fun` es la función de $\mathbb{R}$ en $\mathbb{R}$ a ser integrada, $a, b \in \mathbb{R}$ son los extremos de integración, N es la cantidad de subintervalos a usar y `regla` es un string, que deberá ser `trapecio`, `pm` o `simpson`. La salida S debe ser un número real.

2. Ejecutar los comandos necesarios para mostrar en pantalla los errores absolutos de integrar numéricamente
$$\int_0^1 e^{-x}\,dx,$$
   usando 4, 10 y 20 subintervalos con las 3 reglas compuestas del ejercicio 1.

3. Escribir una función en `python` llamada `senint` que para cada $x \in \mathbb{R}^n$ retorne $y \in \mathbb{R}^n$ tal que y_i es la aproximación numérica de
$$\int_0^{x_i} \cos(t)\,dt,$$
   usando la regla compuesta del trapecio con N_i subintervalos. La cantidad N_i de subintervalos debe ser escogida de forma que la longitud de los subintervalos sea menor o igual a 0.1 (ver comandos `floor, ceil, round`). Para `x = 0,...,2`π, con pasos de 0.5, grafique simultáneamente `sin(x)` y `senint(x)`.

4. Calcular mediante la regla del trapecio compuesta y la regla de Simpson compuesta, las siguientes integrales, con una tolerancia de error de 10^{-5}:

   a) $I = \displaystyle\int_0^1 x\,e^{-x}\,dx,$

   b) $I = \displaystyle\int_0^1 x\,\mathrm{sen}(x)\,dx,$

   c) $I = \displaystyle\int_0^1 (1+x^2)^{3/2}\,dx,$

5. Calcular las siguientes integrales haciendo uso de la librería `scipy` (explorar la función `integrate.quad`)

   a) $I = \displaystyle\int_{-\infty}^{\infty} e^{-x^2}\,dx,$

   b) $I = \displaystyle\int_0^2 x^2\,log\Big(x + \sqrt{x^2+1}\Big)\,dx.$

6. El período de un péndulo de longitud l con amplitud α puede aproximarse con la fórmula:

$$T = 4\sqrt{\frac{l}{g}} \int_0^{\pi/2} \frac{d\theta}{(1 - sen^2(\frac{\alpha}{2})\, sen^2(\theta))^{\frac{1}{2}}},$$

   donde $g = 9.8m/s^2$

   Programar una función, `pendulo`, que reciba una longitud l en metros y α en forma de un número entero entre 0 y 90, transforme el valor a radianes y devuelva el período del péndulo de longitud l. ¿Qué ocurre en el caso de $\alpha = 0$?

7. Se desea implementar una regla de cuadratura adaptiva, es decir, una cuadratura compuesta que utilice más subintervalos en la zona en que la aproximación obtenida sea peor. Para ello, notamos $S(a, b)$ a la regla de Simpson en el intervalo $[a, b]$. Si notamos $c = \frac{a+b}{2}$, se tiene que:

$$\frac{|S(a, b) - S(a, c) - S(c, b)|}{15} \approx E(a, c, b),$$

   donde $E(a, c, b)$ es el error cometido al aplicar la regla compuesta: $S(a, c) + S(c, b)$. Implementar un programa que reciba como input una función f, un intervalo $[a, b]$ y una tolerancia ϵ y calcule las cuadraturas: $q = S(a, b)$, $q_1 = S(a, c)$ y $q_2 = S(c, b)$. Si $|q - q_1 - q_2| < 15\epsilon$, se devuelve el valor $q_1 + q_2$. En caso contrario, se aplica el mismo criterio para integrar f en los intervalos $[a, c]$ y $[c, b]$, con una tolerancia $\frac{\epsilon}{2}$.

   Probar el programa calculando $\displaystyle\int_0^1 xe^{-x^2}\, dx = \frac{1}{2}(1 - e^{-1})$. Comparar los resultados (y los tiempos de ejecución) con los obtenidos por la regla de Simpson compuesta.