ANÁLISIS NUMÉRICO I/ANÁLISIS NUMÉRICO – 2022
Parcial 1 Laboratorio

Fecha de inicio: 22/04/2021
Fecha de entrega: 25/04/2021 23:59
Forma de entrega:

- Archivos `.py` enviados en la tarea creada en el aula virtual. Agregar todos los archivos necesarios para correr las soluciones desde la carpeta de la entrega.
- Dejar instrucciones de ejecución de cada uno en los comentarios o en un único archivo de texto para todos los `.py`.

1. Implementar una función llamada `serie_seno` que reciba x y calcule los primeros 5 términos de la serie de Taylor del seno alrededor de cero:

$$f(x) = \sum_{n=0}^{4} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

2. Visualizar mediante un gráfico la función f en el intervalo $[0, 6.4]$, para una lista de puntos equidistantes separados por una distancia de 0.01 entre sí (sin utilizar la librería `numPy`).

3. Encontrar las dos raíces positivas con el método de bisección, con una cantidad máxima de 100 iteraciones y una tolerancia de 1e-5 (estimar los intervalos de búsqueda a partir del gráfico). Los argumentos de salida deben ser `(hx,hf)` donde `hx`$= [x_1, \ldots, x_N]$ es una lista que representa el historial de puntos medios y `hf`$= [f(x_1), \ldots, f(x_N)]$ el historial de los respectivos valores funcionales.

4. Modifique el programa del método de Newton para transformarlo en el método de Steffensen (sin derivadas), cuya funcion de iteración es

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}$$

   La función debe llamarse `rsteffensen`, y tener como entrada los argumentos `(fun,x0,err,mit)`, donde `fun` es una función que dado x retorna $f(x)$, `x0` es el punto inicial, `err` es la tolerancia deseada del error y `mit` es el número máximo de iteraciones permitidas. El algoritmo debe finalizar en la k-esima iteración si $|f(x_k)| <$ `err` o si $k \geq$ `mit`. Los argumentos de salida deben ser `(hx,hf)` donde `hx`$= [x_1, \ldots, x_N]$ es una lista que representa el historial de puntos generados y `hf`$= [f(x_1), \ldots, f(x_N)]$ el historial de los respectivos valores funcionales.

5. Encontrar las dos raíces positivas de `serie_seno(x)` comenzando con puntos iniciales 3 y 6, con una cantidad máxima de 100 iteraciones y una tolerancia de 1e-5. ¿Cuántas iteraciones requiere cada búsqueda? ¿Qué ocurre al iniciar la búsqueda en 4.5?