Métodos numéricos - Tarea 2

Giovanni Gamaliel López Padilla

1. **Find the fourth Taylor polynomial $P_4(x)$ for the function $f(x) = xe^{x^2}$ about $x_0 = 0$.**
   Como se sabe que
   $$e^x = \sum_{i=0}^{n} \frac{x^i}{i!}$$
   entonces
   $$\begin{aligned} f(x) &= xe^{x^2} \\ &= x\left(\sum_{i=0}^{n} \frac{(x^2)^i}{i!}\right) \\ &= \sum_{i=0}^{n} x\left(\frac{x^{2i}}{i!}\right) \\ &= \sum_{i=0}^{n} \frac{x^{2i+1}}{i!} \end{aligned}$$
   por lo tanto, la función a implementar es:
   $$f(x) = \sum_{i=0}^{n} \frac{x^{2i+1}}{i!}$$

   a) **Find an upper bound for $|f(x) - P_4(x)|$, for $0 \leq x \leq 0.4$, ie find an upper bound of $|R_4(x)|$ for $0 \leq x \leq 0.4$**
   Los límites superiores que se encontraron para $|f(x) - P_4(x)|$ fue 0.000000 y 0.000011 para $|R_4(x)|$.

   b) **Approximate $\int_0^{0.4} f(x)dx$ using $\int_0^{0.4} P_4(x)dx$**
   El resultado de la integral usando el polinomio $P_4(x)$ es de 0.086784.

   Los resultados anteriores pueden ser verificados en el script contenido en la carpeta Problema_1, los valores del salida del programa son los que se muestran en la figura 1.

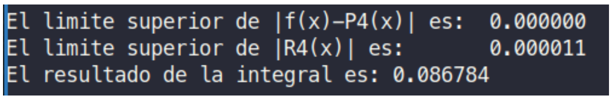

Figura 1: Captura de pantalla de los valores de salida del programa.

2. **Implement a function to comput**
   $$f(x) = \frac{1}{\sqrt{x^2+1} - x}$$
   When evaluating the previous function we can lose accuracy, transform the right hand side to avoid error (or improve the accuracy). Implement the transformed expression and compare the results with the original function. Note: use values of x greater than 10000.

3. **Implement a function to compute the exponential function by using the Taylor/Maclaurin series**

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Since we cannot add infinite terms, we can approximate this expansion by

$$e^x = \sum_{i=0}^{n} \frac{x^i}{i!}$$

El programa de este problema se encuentra contenido en la carpeta Problema_3.

4. **Riemann sums can be used to estimate the area under the curve $y = f(x)$ in the interval $[a, b]$. Left-and right-endpoint approximations, with subintervals of the same width, are special kinds of Riemann sums, ie,**

Left-endpoint approximation:

$$L_n = \sum_{i=1}^{n} f(x_{i-1})\Delta x$$

Right-endpoint approximation:

$$R_n = \sum_{i=1}^{n} f(x_i)\Delta x$$

where $\Delta x = \dfrac{b-a}{n} \qquad x_i = a + i\Delta x,\ i = 0, 1, \cdots, n$

Implement functions to compute $\mathbf{L_n}$ and $\mathbf{R_n}$. Using the function $f(x) = sinx$ over the interval $[0, \frac{\pi}{2}]$, compute $\mathbf{L_n}$ and $\mathbf{R_n}$ for n=10. Compare the previous results with

$$\int_0^{\frac{\pi}{2}} f(x)dx$$

El programa que emplea los dos algoritmos esta contenida en la carpeta Problema_4. Los resultados de cada integral son los que se muestran en la tabla 1 y el programa arroja los resultados en la terminal como se muestra en la figura 2. Con estos resultados podemos obtener la diferencia entre los algoritmos, que es 0.157080.

Algoritmo	**Resultado**
Left-endpoint	0.919403
Right-endpoint	1.076483

Tabla 1: Resultados de los algoritmos de integración con los parámetros especificados para $f(x) = sin(x)$

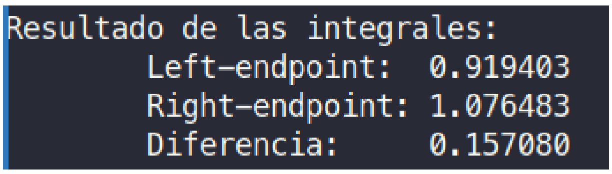

Figura 2: Resultados impresos en la termianl obtenidos por el programa