Taller # 2: Errores

Elaborado por Denis Perez / Heyzel Moncada

2 de Diciembre de 2019

Pregunta 1.a

En una misma ventana grafique la función $f(x)$ con $x \in [0, 1]$ para los siguientes valores de $n = 5, 10, 20, 30, 40, 50, 60$ ¿Qué ocurre con el área bajo la curva de $f(x)$ a medida que n aumenta?

Códigos

La implementación que responde la pregunta planteada es `ejercicio1a.py`

El área bajo la curva de $f(x)$ en $[0, 1]$ sera menor a medida que n aumente. Esto no es mas que una consecuencia de los sucesivos errores de calculo generado tras varias iteraciones.

Pregunta 1.b

Demuestre que $y_n = \int_0^1 \frac{x^n}{x+5}$ satisface la relación $y_n = \frac{1}{n} - 5y_{n-1}$. Para $n \geq 1$ con $y_0 = \int_0^1 \frac{1}{x+5} = log6 - log5 = log6/5$. ¿Qué valores usted espera para y_n $n \to \infty$?

Demostraremos que $y_n + 5y_{n-1} = \frac{1}{n}$

$$\int_0^1 \frac{x^n}{x+5} \quad + \quad \int_0^1 \frac{5x^{n-1}}{x+5} \quad = \quad \frac{1}{n}$$

$$\int_0^1 \frac{x^n + 5x^{n-1}}{x+5} dx \quad = \quad \int_0^1 \frac{x^{n-1}(x+5)}{x+5} dx$$

$$\int_0^1 x^{n-1} dx \quad = \quad \frac{x^n}{n}\Big|_1^0$$

$$\frac{1}{n}((1^n) - (0^n)) = \frac{1}{n}$$

De esta forma logramos demostrar que

$$y_n = \int_0^1 \frac{x^n}{x+5}$$

satisface la relación

$$y_n = \frac{1}{n} - 5y_{n-1}$$

Se espera que los valores de y_n sean cada vez mas pequeños cuando $n \to \infty$

Pregunta 1.c

Pruebe su rutina con los mismos valores de n del ítem **(a)**. ¿El valor de y_n arrojado por su rutina se corresponde con el resultado observado en el ítem **(a)**? Más aún, ¿El valor de y_n arrojado por su rutina se corresponde con el resultado teórico probado en el ítem **(b)**?

Códigos

El código que da respuesta a la pregunta planteada es `ejercicio1c.py`

Los valores de y_n no corresponden en absoluto con los resultados del ítem(a) y tampoco se corresponde al resultado teórico probado en el ítem(b)

Pregunta 1.d

Encuentre una expresión que indique como se propaga el error inicial en el valor inicial de y_0. ¿Esta expresión le permite entender el comportamiento del Algoritmo anterior?

Sea $\vartheta(y_{p-1})$ el error de propagación del termino y_{p-1} de la relación de recurrencia $y_n = \frac{1}{n} - 5y_{n-1}$, Una expresión del error para y_n viene dada por

$$E_{y_n} = \frac{1}{n}\vartheta(y_{n-1}) + 5(y_{n-1}\vartheta(y_{n-1}))$$

$$= \frac{1}{n}\vartheta(y_{n-1}) + \frac{5\vartheta(y_{n-2})}{n-1} + 25(\frac{1}{n-2} - 5y_{n-3})$$

$$\vdots$$

$$E_{y_n} = \sum_{i=0}^{n-1} \frac{5^i}{n-i}\vartheta(y_{n-1})$$

La expresión anterior permite visualizar de manera clara que el error de propagación incrementa mientras $n \to \infty$.

Pregunta 2.a

Independientemente del valor de n, ¿Cúal es el mensaje que debería imprimir el programa?

El mensaje que **deberia** imprimir el programa es: `Calculé la norma 1 de` x

Pregunta 2.b

Programe y ejecute el Algoritmo 2 con $n = 5$ y $n = 10$. Comente las salidas del programa y explique su comportamiento.

Códigos

El código que da respuesta a la pregunta planteada es `ejercicio2b.py`

Para $n = 5$ la salida del programa es `Calculé la norma 1 de x` mientras que para $n = 10$ es `Algo está extraño`. Esto se debe a que la representación del número 0.2 en binario es un número punto flotante con mantisa infinita, un computador evidentemente no puede representar un número con mantisa infinita, así que lo que hace es truncar el número, entonces al hacer el calculo se tendrá una aproximación de la norma 1 de x pero no la norma 1 de x exactamente.

Pregunta 3.a

¿Cuál es el resultado de Ax SIN el computador?

$$Ax = \begin{pmatrix} 0.4 & -0.6 & 0.2 \\ -0.3 & 0.7 & -0.4 \\ -0.1 & -0.4 & 0.5 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Pregunta 3.b

¿Qué espera que ocurra en la primera iteracion, para la A y el x dados?

Se espera que ocurra un error del tipo NaN debido a una división entre 0.

Pregunta 3.c

Programe *Potencia* y ejecútela. ¿Ocurrió lo esperado? Explique el comportamiento de la rutina.

Códigos

El código que da solución a la pregunta es `ejercicio3c.py`

No, no ocurrió lo esperado. El algoritmo de iteración de potencia comienza con un vector b_0, que puede ser una aproximación al vector propio dominante o un vector aleatorio. El método se describe mediante una relación de recurrencia

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}$$

Entonces, en cada iteración, el vector b_k se multiplica por la matriz A y es normalizado. Finalmente el algoritmo realiza un cociente denominado *Cociente de Rayleigh* para dar con el autovalor máximo de la matriz.

Pregunta 3.d

¿Cuál es la funcionalidad de la rutina Potencia?

Códigos

En esta cajita coloque el nombre de los archivos `ejercicio3d.py`

El método de las potencias es un método iterativo que calcula sucesivas aproximaciones a los autovectores y autovalores de una matriz. Se usa principalmente para aproximar el autovector dominante de mayor autovalor en matrices grandes. El nombre podría venir de el hecho de tener que multiplicar una matriz A sucesivamente por un vector x y luego normalizarlo dejando de alguna forma una expresión matemática que se puede asemejar a la siguiente:

$$x_n = \frac{A^n x}{\|Ab_k\|}$$

Pregunta 3.e

¿Puede afirmar que los errores numéricos son siempre indeseables?

Los errores numéricos no son siempre indeseables, mas aún, de no haber errores numéricos el método de las potencias no funcionaria y dicho algoritmo es sumamente utilizado debido a sus increíbles resultados.

Pregunta 4.a

Demuestre que

$$\lim_{x\to 0} f(x) = \frac{1}{6}$$

$$lim_{x\to 0}\frac{x - sin(x)}{x^3}$$

$$= \lim_{x\to 0}\frac{1 - cos(x)}{3x^2} \quad \text{(L'Hopital)}$$

$$= \lim_{x\to 0}\frac{sin(x)}{6x} \quad \text{(L'Hopital)}$$

$$= \lim_{x\to 0}\frac{cos(x)}{6} \quad \text{(L'Hopital)}$$

$$= \frac{cos(0)}{6} = \frac{1}{6} \quad \text{(Sustitución)}$$

Pregunta 4.b

Escriba una rutina que evalúe $f(x)$ 15 veces: La primera vez en $x = 1$ y las restantes veces en x := $\frac{1}{10x}$. Observe que $x \to 0$. ¿Los resultados de su evaluación coinciden con lo probado en el ítem **(a)**?

Códigos

El código que da respuesta a la pregunta planteada es `ejercicio4b.py`

Los resultados no coinciden con lo probado en el ítem (a). Esto se debe a que no se esta evaluando $f(x)$ con los valores exactos, si no con aproximaciones a esos valores, entonces mientras más cercana es la aproximación, más cercano es el resultado. Sin embargo, a partir de cierto valor el computador no puede representar bien la aproximación ya que es un número muy pequeño y da resultados que no tienen ninguna semejanza con el esperado.

Pregunta 4.c

Demuestre que el siguiente pseudo código genera una aproximación a $f(x)$ alrededor de $x = 0$

De dicho pseudo código, suponiendo el caso particular $n = 3$ obtenemos el siguiente polinomio:

$$\frac{x^6}{1995840} - \frac{x^4}{18144} + \frac{x^2}{252} + \frac{1}{6}$$

que al evaluar x alrededor de 0, da como resultado un número aproximado a $\frac{1}{6}$, entonces probemos con un $n = k$

$$\frac{1}{6} + \frac{x^2}{252} - \frac{x^4}{18144} + \frac{x^6}{1995840} - \cdots + (-1)^{k-1}\frac{x^{m_1}}{C_1} + (-1)^k\frac{x^{m_2}}{C_2}$$

Donde m_1,m_2 son de la forma $2h$ con $h \in \mathbb{Z}$ y $C_1 < C_2$. Ahora bien, sabemos que la expresión desarrollada anteriormente genera un polinomio por lo que podríamos encontrar el punto de corte con el $ejey$ para saber si en verdad estamos generando una aproximación.

$$y(0) = \frac{1}{6}$$

De modo que si el polinomio tiene un punto de corte en $(0, \frac{1}{6})$ entonces estaremos generando una aproximación de el valor real de $f(x)$, mas aún, si tomamos el limite de la expresión anterior

$$\lim_{x\to 0} \frac{1}{6} + \frac{x^2}{252} - \frac{x^4}{18144} + \frac{x^6}{1995840} - \cdots + (-1)^{k-1}\frac{x^{m_1}}{C_1} + (-1)^k\frac{x^{m_2}}{C_2} = \frac{1}{6}$$

Entonces estamos comprobando que efectivamente cuando tomamos valores cercanos a 0 con cualquier cantidad de términos para nuestro pseudo código éste tiende hacia $\frac{1}{6}$, por lo tanto, se concluye que efectivamente el pseudo código propuesto genera una aproximación de $f(x)$ en el punto $x = 0$

Pregunta 4.d

Programe y use la rutina anterior para aproximar $f(x)$ en los valores de x usados en el ítem **(b)**. Comente y justifique los resultados. ¿Cuál forma de evaluar $f(x)$ usted

considera más precisa?

Códigos

El código que da respuesta a la pregunta planteada es `ejercicio4d.py`

La rutina en el apartado **(b)** presenta una mejor aproximación de $f(x)$ en sucesivas iteraciones hasta llegar al punto en donde no puede aproximar mas debido a las limitaciones de aritmética finita del computador. Por otra parte la rutina del apartado **(c)** presenta una mejor estabilidad con una peor aproximación (pero aun aceptable). Tomando en cuenta la estabilidad de ambas rutinas se puede apreciar claramente que la rutina del apartado **(b)** es mas sensible a los cambios en la data de entrada por lo que su estabilidad es poca, sin embargo, la rutina del apartado **(c)** presenta una mejor estabilidad pudiendo hacerse cambios en la data de entrada y seguir teniendo una data de salida lo suficientemente estable.