Análisis numérico

Facultad de Ciencias, UNAM
Semestre 2022-1
Práctica Python

Profesor: César Carreón Otañez
Ayudante: Isaí López Servín
Ayudante: Jorge Zavaleta Sánchez

Fecha de entrega: *29 de octubre de 2021*

Indicaciones generales

Todos los ejercicios deben ser realizados usando `Python`. **Es obligatorio que para realizar su práctica considere lo siguiente**:

- Puede entregar los ejercicios en una libreta de `Jupyter` o `Google Colab` y entregar un sólo archivo `.ipynb` nombrado `PPython_` y después del guión bajo ponga las iniciales de los apellidos de los integrantes del equipo en orden alfabético, por ejemplo, `PPython_CLZ.ipynb`.

- También pueden entregar sus ejercicios mediante archivos `.py`, al menos uno por cada ejercicio. Nombre el archivo principal como `E#.py` donde `#` debe ser remplazado por el número del ejercicio. Para enviar su práctica deben comprimir sus archivos en un solo fichero de su gusto (`.rar`, `.zip`, `.7z`, `.tar`, `.tar.gz`, etc.). Para el nombre del archivo comprimido use las indicaciones del punto anterior, por ejemplo, `PPython_CLZ.zip`.

- Todos los archivos `.py` o `.ipynb` que entreguen, deben contener la información de los autores.

- Es necesario que el código este brevemente comentado utilizando los comentarios propios de `Python` o en celdas `Markdown` si se usan las libretas. (Vea pág. 5)

Notas importantes

- **La práctica la debe de entregar un sólo miembro del equipo**.

- **El archivo debe ser entregado únicamente mediante Classroom**. Es responsabilidad del alumno verificar que el archivo entregado sea el correcto, se pueda compilar o ejecutar y de que se subió de forma adecuada a *Classroom*.

- **Cada ejercicio, al inicio del enunciado, tiene marcado su puntaje**. La calificación máxima de la práctica es de 100 puntos.

Ejercicios

E1. [10pts.] El valor de π puede ser estimado a partir de la expresión

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdots$$

Escriba un `script` que determine π para cualquier número de términos. Para hacer esto, se debe preguntar al usuario el número de términos, después calcular el correspondiente valor de la aproximación de π y mostrarla al usuario.

E2. [10pts.] Un *número primo de Mersenne* es un número primo que es igual a $2^n - 1$, donde n es un entero. Por ejemplo, 31 es un número primo de Mersenne ya que $31 = 2^5 - 1$. Escriba un `script` que encuentre todos los números primos de Mersenne entre 1 y 10,000.

E3. [15pts.] La expasión en series de Taylor para $\sin x$ alrededor de $x = 0$ esta dada por

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}.$$

donde el ángulo x está en radianes.

a) Haga una función que determine $\sin x$ usando la expasión en series de Taylor. Como nombre de la función use `sinTay` y como argumento de entrada use `x`, de tal manera que el llamado a la función sea `sinTay(x)`. El argumento de entrada `x` es el ángulo en grados y la función debe regresar el valor de $\sin$ `x`. Dentro de la función debe usar un ciclo para agregar términos de la serie. Si a_k es el k-ésimo término de la serie, entonces la suma parcial S_k de los k términos es $S_k = S_{k-1} + a_k$. En cada paso calcule el estimado del error E dado por

$$E = \left| \frac{S_k - S_{k-1}}{S_{k-1}} \right|.$$

Pare de sumar términos cuando $E \leq 0.000001$. Dado que $\sin\theta = \sin(\theta \pm 360n)$ (n entero) escriba su función de tal manera que si el ángulo es mayor que $360°$ o menor que $-360°$, entonces la serie de Taylor sea calculada usando el menor número de términos (usando un valor para x que sea más cercano a 0).

b) Haga un `script` donde reporte los resultados obtenidos por su función `sinTay` al calcular

1) $\sin(39°)$.
2) $\sin(205°)$.
3) $\sin(-70°)$.
4) $\sin(754°)$.
5) $\sin(19000°)$.
6) $\sin(-748°)$.

y compare sus resultados con los que se obtienen al hacer el cálculo directamente en `Python` con la funciones `sin` implementadas en los módulos `math` o `numpy`.

c) Realice la gráfica de su función `sinTay` en el intervalo $[-360°, 360°]$ y compárela con la gráfica de la función `sin` evaluada en el mismo intervalo. La comparación debe ser hecha en la misma figura.

E4. [20pts.] Haga funciones en `Python` que implementen las siguientes operaciones con matrices utilizando solamente ciclos (`for` ó `while`) y referencias* a los elementos de las matrices. Para representar las matrices en `Python` puede usar listas, tuplas o arreglos de `numpy`, sólo que para la último opción no se pueden usar los operadores ya implementados para los `ndarray`.

a) *Suma de matrices*. Sean $A, B, C \in \mathcal{M}_{m\times n}(\mathbb{R})$**, la suma de matrices $C = A + B$ tiene por entradas $c_{i,j} = a_{i,j} + b_{i,j}$ para $1 \leq i \leq m$, $1 \leq j \leq n$.

b) *Producto de matrices*. Sean $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, $B \in \mathcal{M}_{n\times p}(\mathbb{R})$ y $C \in \mathcal{M}_{m\times p}(\mathbb{R})$, el producto de matrices $C = AB$ tiene por entradas $c_{i,j} = \displaystyle\sum_{k=1}^{n} a_{i,k}b_{k,j}$ para $1 \leq i \leq m$, $1 \leq j \leq p$.

c) *Traza de una matriz*. Sea $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, la traza se define como $\operatorname{tr}(A) = \displaystyle\sum_{i=1}^{n} a_{i,i}$.

d) *Transpuesta de una matriz*. Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, la transpuesta de A, la cual se denota como $A^\intercal \in \mathcal{M}_{n\times m}(\mathbb{R})$ tiene por entradas $(A^\intercal)_{i,j} = a_{j,i}$ para $1 \leq i \leq m$, $1 \leq j \leq n$.

e) Utilice las funciones creadas en los incisos anteriores y las matrices

$$A = \begin{pmatrix} 17 & -3 & 6 & 13 & 12 \\ -8 & 5 & 15 & -9 & 10 \\ -7 & 7 & -20 & 3 & -22 \\ -22 & 2 & -7 & 7 & 10 \\ -7 & 1 & 5 & -9 & -2 \end{pmatrix} \qquad B = \begin{pmatrix} -9 & 9 & 3 \\ -10 & 1 & 1 \\ -23 & -10 & -9 \\ 7 & -5 & -6 \\ -3 & -8 & -27 \end{pmatrix}$$

$$C = \begin{pmatrix} -8 & 2 & -2 & 8 & -10 \\ -1 & 4 & 16 & 8 & 11 \\ -5 & -16 & 9 & -15 & -18 \end{pmatrix} \qquad D = \begin{pmatrix} 13 & 24 & -11 \\ 1 & 1 & -17 \\ 16 & -2 & -10 \end{pmatrix}$$

para realizar las siguientes operaciones escritas en un `script`

- $A + BC$
- $C + B^\intercal$
- ABC
- $BD + C^\intercal$
- $(BDC + A)^\intercal$
- $\operatorname{tr}((CB)^\intercal + D)$

*Dada la matriz A si queremos hacer referencia a la entrada i, j de A (denotada por $a_{i,j}$), recuerde que en `Python` se usa la notación `A[i,j]` con los arreglos de `numpy` o `A[i][j]` si se usan listas o tuplas.

**La notación $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ significa que A es una matriz con m filas y n columnas y sus entradas $a_{i,j}$ son reales para $i = 1, \ldots, m$ y $j = 1, \ldots, n$.

E5. [20pts.] Una transformación $w\colon \mathbb{R}^2 \to \mathbb{R}^2$ de la forma

$$w(\overline{x}) = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

donde a, b, c, d, e y f son números reales, y $\overline{x} = (x, y)^\intercal \in \mathbb{R}^2$ es llamada una transformación afín (de dimensión 2). Esta transformación afín se puede representar en forma matricial, esto es

$$w(\overline{x}) = A\overline{x} + \overline{b}$$

para

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \overline{b} = \begin{pmatrix} e \\ f \end{pmatrix}.$$

El fractal del helecho (*helecho Barnsley*) puede ser implementado al graficar puntos creados iterativamente de acuerdo a la transformación afín

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = w_H(\overline{x}_k) = A \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \overline{b}$$

para A y $\overline{b}$ elegidos en cada iteración k de manera aleatoria con diferente probabilidad de entre una de las siguientes reglas

Regla 1 (con 85 % de probabilidad)

$$A = \begin{pmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{pmatrix} \quad \overline{b} = \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

Regla 2 (con 7 % de probabilidad)

$$A = \begin{pmatrix} 0.20 & -0.26 \\ 0.23 & 0.22 \end{pmatrix} \quad \overline{b} = \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

Regla 3 (con 7 % de probabilidad)

$$A = \begin{pmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{pmatrix} \quad \overline{b} = \begin{pmatrix} 0 \\ 0.44 \end{pmatrix}$$

Regla 4 (con 1 % de probabilidad)

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0.16 \end{pmatrix} \quad \overline{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

a) Escriba una función que regrese dos vectores `x` y `y` cuyas entradas `k` (`k = 0` hasta `n-1`) correspondan a los `n` puntos generados iterativamente por la transformación w_H. Tome como punto inicial `x`$[0] = x_0 = 0.5$ y `y`$[0] = y_0 = 0.5$.

b) Use un script y la función `subplot` para mostrar las gráficas que se obtienen con `n` $=$ 50, 500, 5,000 y 50,000 iteraciones (puntos). Para generar las gráficas, use la función del inciso anterior y grafique `x` vs `y` como puntos individuales (use `scatter`).

E6. [25pts.] Realice un programa que permita calcular la edad de una persona (al día de hoy) a partir de su fecha de nacimiento. Para ello, pida la fecha de nacimiento del usuario (`dd-mm-aaaa`) y haga una o varias funciones que permitan calcular la edad en años, meses y días. Por ejemplo, si la fecha de nacimiento es `22-4-2000`, la respuesta del programa debe ser 21 años, 6 meses y 7 días considerando que la fecha de hoy es la fecha de entrega (`29-10-2021`).

El programa debe comprobar que la fecha de nacimiento sea válida (es decir, que exista en el calendario), que no sea una fecha mayor que la fecha del día de hoy y en dado caso que no se cumplan estas condiciones mandar los mensajes correspondientes. Puede usar el módulo `datetime` para recuperar la fecha de hoy de su sistema mediante el siguiente código:

```
from datetime import date
hoy = date.today()
print(hoy.day,hoy.month,hoy.year)
```

Nota: No puede usar ninguna otra función o estructura del módulo `datetime` aparte de las líneas previas.

Anotaciones

Este debe ser el formato a seguir para los archivos `.py` usando comentarios

```
# Nombre(s) del autor(es)
# Descripcion del ejercicio
# Si es una funcion se debe especificar lo siguiente:
  # Breve descripcion de la rutina
  # Argumentos de entrada: tipo de argumento
  # Argumentos de salida: tipo de argumento
  # Dependencia de otras rutinas
```

En las libretas `.ipynb` debe usar el mismo formato en una celda de código o adecuarlo a una celda `markdown`.