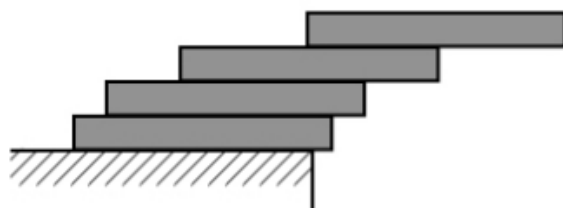


1 (Problema de apilamiento de libros) Aquí hay una situación física interesante en la que aparece la Serie Armónica. El problema fue publicado por Paul Johnson en 1955. El código `harmonic.ipynb` puede ser útil en esta pregunta.

Deseo crear una torre inclinada de libros utilizando múltiples copias idénticas de un libro. Usando n libros organizados en una torre perpendicular al borde de la mesa, empujo el libro superior tan lejos como puedo, y hago lo mismo con el siguiente libro abajo, trabajando hacia la base de la torre. Ver la figura abajo cuando $n = 4$. Podemos asumir que los libros y la mesa son lo suficientemente rígidos como para que no haya movimientos verticales.



- (a) Demuestra que utilizando n libros, el voladizo (en unidades de libros) se puede escribir como la Serie Armónica

$$\frac{1}{2} \left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \right).$$

Deduce que usando 4 libros, el voladizo excede la longitud de un libro.

- (b) Grafica el voladizo (en unidades de libros) en función del número de libros usados para construir la torre. Considera hasta 1000 libros. Sugerencia: usa escala logarítmica en el eje horizontal.
- (c) En la misma gráfica, traza el resultado cuando se usa la ecuación 1.4 (aproximación logarítmica de la Serie Armónica) para calcular el voladizo.
- (d) Usando la aproximación logarítmica:
- estima el voladizo cuando se usan 10^6 libros para crear la torre. (Respuesta: voladizo de alrededor de 7 libros de longitud.)
 - estima el número de libros necesarios para construir una torre inclinada con un voladizo de 10 libros de longitud. (Tu respuesta probablemente sea mayor que el número estimado de libros físicos que existen en el mundo.)

2 (Aproximaciones famosas para π) A continuación se presentan tres aproximaciones históricamente importantes para π .

- **Serie de Madhava** (siglo XIV), a veces llamada la *aproximación de Gregory-Leibniz* (1671-1673)

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \right)$$

- **Producto de Wallis** (1656)

$$\pi = 2 \left(\frac{2}{1} \cdot \frac{2}{3} \right) \cdot \left(\frac{4}{3} \cdot \frac{4}{5} \right) \cdot \left(\frac{6}{5} \cdot \frac{6}{7} \right) \cdots$$

- **Fórmula de Viète** (1593)

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

Sea n el número de iteraciones en cada esquema de aproximación. Por ejemplo, la iteración cero ($n = 0$) da 4, 2 y 2 para las tres aproximaciones respectivamente.

- (a) En el mismo conjunto de ejes, grafica los resultados de las tres aproximaciones en función del número de iteraciones hasta $n = 10$.
- (b) En un conjunto de ejes logarítmicos, grafica el error fraccional absoluto

$$\left| \frac{\text{Estimación después de } n \text{ iteraciones} - \pi}{\pi} \right|$$

para las tres aproximaciones hasta $n = 100$. Esto nos da una idea de qué tan rápido las aproximaciones convergen a π .

Deberías encontrar que el error para la fórmula de Viète no parece disminuir por debajo de un límite mínimo de alrededor de 10^{-16} .

- (c) Recuerda que una estimación x de π es precisa hasta p lugares decimales si $|x - \pi| < 0.5 \times 10^{-p}$. Para cada una de las tres aproximaciones de π , calcula cuántas iteraciones se necesitan para obtener π preciso hasta 5 lugares decimales. (Respuestas: 200000, 157080 y 9.)

3 (Fórmula de Ramanujan para π) En 1910, Ramanujan dio la siguiente fórmula para π .

$$\pi = \left[\frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^{4n}} \right]^{-1}.$$

(Famosamente, él simplemente "escribió" muchas de estas fórmulas.) Calcula las primeras 3 iteraciones de esta aproximación. ¿A cuántos lugares decimales es precisa cada una de estas aproximaciones? Intenta escribir un código que calcule la serie hasta n términos. ¿Puede tu código evaluar el resultado con precisión, por ejemplo, cuando $n = 10$? Si no, explica por qué.

4 (Número recíproco de Fibonacci) Usa `fibonacci.ipynb` como punto de partida.

La constante recíproca de Fibonacci está dada por

$$\psi = \sum_{n=1}^{\infty} \frac{1}{F_n} = 3.35988566624317755 \dots$$

- (a) Calcula la suma hasta 20 términos. Sugerencia: Intenta hacer esto de manera vectorizada usando arreglos en lugar de un bucle.
- (b) Calcula cuántos términos son necesarios para obtener ψ con 10 decimales de precisión. (Respuesta: 49.)

5 (Secuencias generalizadas de Fibonacci) Usa `fibonacci.ipynb` como punto de partida.

- (a) Supongamos que usamos diferentes valores iniciales F_0 y F_1 . Usa Python para investigar el comportamiento de la razón $R_n := \frac{F_n}{F_{n-1}}$. Muestra que R_n siempre converge al número áureo. (Si te gusta un desafío, podrías demostrar este comportamiento usando deslizadores.)
- (b) (*Secuencias de Lucas*) Sean P y Q enteros fijos (no ambos cero). Define la secuencia de Lucas, $U_n(P, Q)$, por las siguientes reglas:

$$U_0 = 0, \quad U_1 = 1, \quad U_n = PU_{n-1} - QU_{n-2}.$$

Nota que la secuencia de Fibonacci corresponde a $U_n(1, -1)$. Escribe un código que grafique el comportamiento para n grande de la razón de términos consecutivos $R_n := \frac{U_{n+1}}{U_n}$ para los siguientes valores de P y Q :

- i. $(P, Q) = (3, 1)$
- ii. $(P, Q) = (2, 1)$
- iii. $(P, Q) = (1, 1)$

Haz una conjetura sobre los valores del rango de P tales que R_n sea una secuencia convergente. (Respuesta: ver <https://mathworld.wolfram.com/LucasSequence.html>)

6 (Orden de convergencia) Dada una secuencia convergente, es natural preguntar: *¿qué tan rápido converge la secuencia?* En esta pregunta, exploramos cómo cuantificar la velocidad de convergencia. Supongamos que una secuencia (x_n) converge a x . Define el error absoluto, E_n , como la secuencia

$$E_n = |x_n - x|.$$

La velocidad de convergencia se puede cuantificar mediante dos constantes positivas: la *tasa* (C) y el *orden* (q) de convergencia, definido a través de la ecuación

$$\lim_{n \rightarrow \infty} \frac{E_{n+1}}{(E_n)^q} = C > 0.$$

- (a) Verifica manualmente que la secuencia $(\frac{1}{n})$ converge con $q = C = 1$.
- (b) Para la mayoría de las aplicaciones, solo el orden de convergencia es de interés, ya que es un mejor indicador de qué tan rápido converge la secuencia. Se puede demostrar que podemos estimar q usando la fórmula

$$q \approx \frac{\ln(E_{n+1}/E_n)}{\ln(E_n/E_{n-1})},$$

donde n debería ser lo suficientemente grande como para que q se estabilice, pero no tan grande como para que el denominador en la fórmula sea cero.

- i. Usando esta aproximación, verifica (usando Python) que la razón de números de Fibonacci $R_n = \frac{F_{n+1}}{F_n}$ converge con un orden menor que 1. Sugerencia: podría ser útil graficar q como una función de n . Deberías encontrar que la gráfica se estabiliza en $q = 1$ antes de volverse indefinida cuando n es demasiado grande.
- ii. Considera la serie de Madhava para π en la Pregunta 2. Muestra que $q < 1$. Técnicamente, decimos que la convergencia de la serie es sublineal (es decir, terriblemente lenta).
- iii. Conjetura una secuencia que converge superlinealmente ($q > 1$) y demuestra gráficamente esta propiedad.

7 En `continuityslider.ipynb` (§1.7), agrega un deslizador en la interfaz gráfica de usuario (GUI) para que el valor de x_0 se pueda cambiar.

8 (El huerto de Euclides) La función de Thomae (§1.8) tiene una conexión interesante con el siguiente problema, a veces llamado el huerto de Euclides o el problema de los puntos visibles.

Considera puntos de una cuadrícula con coordenadas enteras positivas (m, n) . Se dice que un punto de la cuadrícula P es visible desde $(0, 0)$ si una línea recta que une $(0, 0)$ con P no pasa a través de ningún otro punto de la cuadrícula. Por ejemplo, como se muestra en la figura de abajo, el punto $(1, 2)$ es visible desde $(0, 0)$, pero $(2, 4)$ no lo es.

Primero deberías intentar experimentar en un trozo de papel cuadriculado para formular una conjetura sobre qué puntos son visibles desde $(0, 0)$.

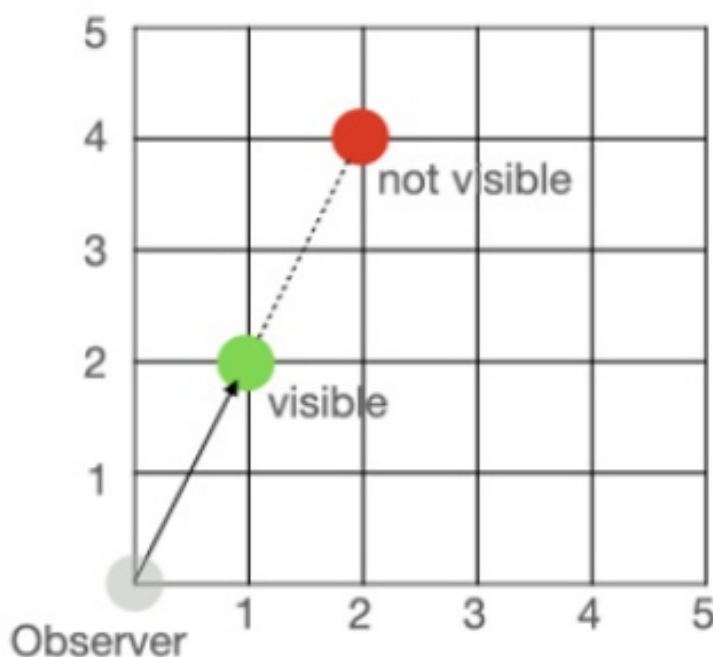
- (a) Escribe un código que produzca una cuadrícula y marque con un punto rojo cada punto visible desde $(0, 0)$ (deberás imponer un límite razonable).
- (b) Para cada punto visible (m, n) , grafica su imagen bajo la siguiente aplicación

$$F(m, n) = \left(\frac{m}{m+n}, \frac{1}{m+n} \right).$$

¿Qué observas? ¿Puedes explicar cómo funciona esta transformación? (Respuesta: ¡Ves la función de Thomae!)

9 (Búsqueda de raíces) Usa el código de bisección para resolver los siguientes problemas. Da tus respuestas con 4 decimales.

- (a) Resuelve la ecuación $x^3 - x^2 - x - 1 = 0$. Sugerencia: Comienza graficando.
- (b) Resuelve $\frac{\sin x}{x} = 0.9$ (verificando así el punto de intersección visto en la fig. 1.8).



- (c) Encuentra el valor numérico de $\sqrt{3}$ usando solo las cuatro operaciones básicas $+$, $-$, \times , \div . Sugereencia: Comienza definiendo $f(x) = x^2 - 3$.

10 (Razón Áurea Generalizada) Supongamos que generalizamos la Razón Áurea a ϕ_n , definida como la raíz positiva del siguiente polinomio de orden n

$$x^n - x^{n-1} - x^{n-2} - \dots - 1 = 0.$$

Por ejemplo, $\phi_1 = 1$ y $\phi_2 = 1.618$. Escribe un código que grafique la secuencia ϕ_n (obtenida mediante bisección). Haz una conjetura sobre el valor de $\lim_{n \rightarrow \infty} \phi_n$.

11 (Método de Newton-Raphson) Sea f una función diferenciable en \mathbb{R} . El método de Newton-Raphson para encontrar la raíz de la ecuación $f(x) = 0$ comprende los siguientes pasos.

- Comienza con una suposición inicial x_0 .
- Calcula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

donde la expresión para $f'(x)$ debe ser conocida explícitamente.

- Repite la iteración anterior mientras $|x_{n+1} - x_n| < \epsilon$ para una tolerancia fija ϵ especificada por el usuario (por ejemplo, 0.5×10^{-p} para una precisión de p decimales). Este paso asegura que solo detengamos el proceso cuando los primeros p decimales estén estabilizados.
- (a) Escribe un código en Python para este método y úsalo para resolver los problemas de la pregunta 9.
- (b) Sea x_n la estimación de la solución de $f(x) = x^2 - 3$ después de n iteraciones usando el método de Newton-Raphson. Calcula el error absoluto $E_n = |x_n - \sqrt{3}|$ y verifica que el método de Newton-Raphson tiene orden 2. Haz esto calculando q definido en la pregunta 6.

12 (Creando una gráfica bonita) Grafica la función *sinc* $f : [0, 2\pi] \rightarrow \mathbb{R}$ definida por

$$f(x) = \begin{cases} \frac{\sin x}{x} & \text{si } x \neq 0, \\ 1 & \text{si } x = 0. \end{cases}$$

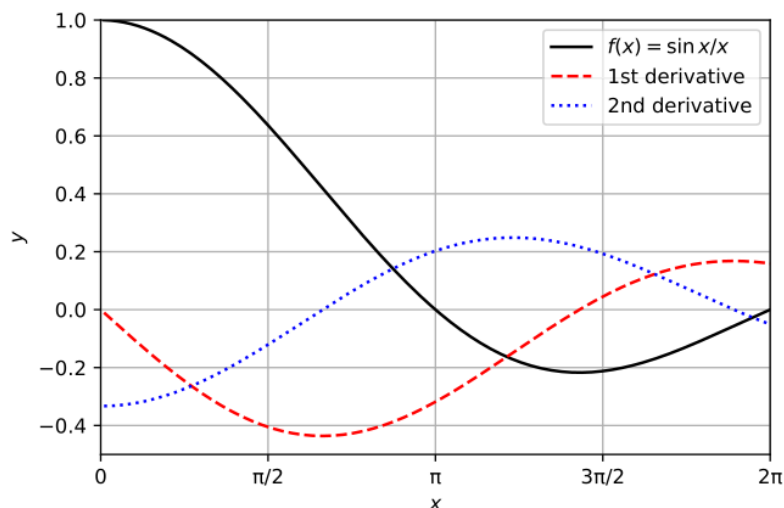


Figure 1: Una bonita gráfica de la función *sinc* y sus primeras y segundas derivadas.

En el mismo conjunto de ejes, grafica su primera y segunda derivada, estimadas usando el método de Euler hacia adelante (es decir, no diferencias nada a mano en esta pregunta). Intenta usar diferentes tipos de líneas en tu gráfica (piensa en los lectores daltónicos). Aquí tienes un ejemplo de salida.

13 (Teorema del Valor Medio) Considera la función *sinc* f definida en la pregunta anterior. Apliquemos el Teorema del Valor Medio a f en el intervalo $[0, \pi]$. En particular, determinemos el(los) valor(es) de c en la declaración del teorema. En otras palabras, deseamos encontrar $c \in [0, \pi]$ tal que

$$f'(c) = \frac{f(\pi) - f(0)}{\pi}.$$

- Demuestra (a mano) que una solución es $c = \pi$.
- Usa un algoritmo de búsqueda de raíces para encontrar otra solución con 4 decimales. Tu respuesta debe ser consistente con la línea discontinua en la fig. 1 anterior.

14 (Otro contraejemplo) Aquí hay otro ejemplo interesante e intuitivo en el análisis. Usa `counterexample.ipynb` como plantilla.

Considera la función $f : [0, 1] \rightarrow \mathbb{R}$ definida por

$$f(x) = \begin{cases} x(2 - \cos \ln x - \sin \ln x) & \text{si } x \in (0, 1], \\ 0 & \text{si } x = 0. \end{cases}$$

Se puede demostrar que f es estrictamente creciente, sin embargo, hay infinitos puntos donde $f'(x) = 0$ en $(0, 1]$.

- Usa Python para graficar la función. Tu gráfica debe demostrar la estructura autosemejante cuando hacemos zoom hacia el origen.
- Demuestra (a mano) que hay puntos de inflexión en $x = e^{-2n\pi}$ donde $n \in \mathbb{N}$. Indícalos en tu gráfica.