

Tarea programada 1

Aunque se puede usar el lenguaje de programación que usted desee como en las otras tareas, se recomienda fuertemente el uso de Python. Al igual que en tareas anteriores, usted debe de escribir un comentario sobre cómo funciona su solución, y debe entregar el código (puede ser un Jupyter notebook) y un pdf (puede ser el mismo Jupyter notebook pero exportado en pdf pero que contenga la salida de ejecución del código como uo lo corrió).

Problema 1

Este es un problema guiado sobre la solución numérica de un problema de valores de frontera por medio de matrices. Le servirá para mecánica cuántica. El problema se ve largo pero es en realidad corto y va muy guiado porque esta técnica de solución no la discutimos en clase.

1. Investigue cómo funciona la función de numpy llamada `np.eye(n, k=0)`. Pruebe algunos valores de n y de k .
2. Considere las matrices $(f_{ij}) = \text{np.eye}(4, k=0)$, $(g_{ij}) = \text{np.eye}(4, k=1)$, $(h_{ij}) = \text{np.eye}(4, k=-1)$. En cada caso, para la fila i , ¿cuál columna j no es cero? Dé su respuesta en términos de i .
3. Tenga en cuenta la siguiente información:

La segunda derivada de una función ψ puede aproximarse numéricamente como

$$\frac{d^2\psi}{dx^2} \approx \frac{\psi_{n+1} - 2\psi_n + \psi_{n-1}}{(\Delta x)^2}$$

Imagine que tenemos un vector columna de entradas ψ_n . ¿Qué suma de matrices $H_p = H_1 + H_2 + H_3$ deben multiplicarse por (ψ_n) , es decir, formar $H_p\psi$, para producir el numerador de la segunda derivada aproximada? Cree estas matrices H_i utilizando `np.eye()` con $N = 1001$. No es necesario crear el vector columna (ψ_n) en ningún momento.

4. Cree un array $x = (x_n)$ con $N = 1001$ valores entre -20 y 20. Calcule la distancia Δx entre los elementos de x .
5. Considere la siguiente información:

La ecuación diferencial ordinaria

$$-\frac{1}{2} \cdot \frac{d^2\psi}{dx^2} + \frac{1}{2} x^2 \psi = E \psi$$

forma un problema de autovalores y puede aproximarse numéricamente como

$$-\frac{1}{2} \cdot \frac{\psi_{n+1} - 2\psi_n + \psi_{n-1}}{(\Delta x)^2} + \frac{x_n^2}{2} \psi_n = E_n \psi_n.$$

El lado izquierdo puede escribirse como $H\psi$ donde H es una suma de matrices creadas con la función `np.eye()`.

Escriba la matriz H . No olvide que el término con x_n^2 también debe estar multiplicado por una de esas matrices diagonales creadas con `np.eye()`.

6. Calcule los autovalores E_n y autovectores ψ_n de H por medio de la función `numpy.linalg.eigh(H)`. Note que los autovectores que la función calcula están ordenados por columnas en la matriz, tal que el autovector ψ_n corresponde a la columna n de la matriz devuelta.
7. Verifique que dos autovectores que ud. elija sean ortogonales y calcule su normalización.
8. Grafique los primeros 6 autovectores $\psi_n(x)$ (esto es, el eje horizontal debe ser el array x y el eje vertical debe ser el n -ésimo autovector). Para que la gráfica quede bien, puede usar el n -ésimo autovalor para desplazar los autovectores verticalmente, es decir, grafique $\psi_n(x) + E_n$. Puede además multiplicar el autovector por algún factor > 1 para mejorar la claridad de la gráfica. Elija el intervalo de visualización $x \in [-5, 5]$.

9. Busque en algún libro las autofunciones (expresiones analíticas) del oscilador armónico cuántico (en términos de los polinomios de Hermite). Grafique cualitativamente las primeras 6 (las constantes no importan) y demuestre que los autovectores de la matriz H con la malla de posiciones discreta (x_n) son una buena aproximación a las autofunciones (analíticas) de la ecuación diferencial. Note cómo se tiende un puente entre los siguientes conceptos: un espacio vectorial de dimensiones finitas y funciones como elementos de un espacio vectorial de dimensiones infinitas; las autofunciones y los autovectores; los operadores diferenciales y las matrices. Información adicional: este mismo método matemático puede usarse para resolver otros problemas de mecánica cuántica de forma numérica.

Libro de referencia para este problema: Wiedemann & Ingold (2025) *Numerische Physik mit Python*, Springer. Cap. 5 (en alemán; no tiene que leer nada de ese libro para hacer el ejercicio).

Problema 2

Tome el ejercicio que hicimos en clase sobre los polinomios de Legendre con matrices (<https://www.gandreoliva.org/cursos/ucr/metodos3/eigenfunctions-legendre.html>). Repita el mismo ejercicio pero ahora hágalo con los polinomios de Laguerre en lugar de los de Legendre. Compare la gráfica de los primeros seis polinomios de Laguerre con alguna obtenida por otros medios (libro, internet, Mathematica, Maxima, etc.).

Sugerencias:

- Empiece escribiendo el operador de Laguerre
- Construya la representación matricial del operador de Laguerre
- Cuando grafique, tenga en cuenta:
 - En los polinomios de Laguerre, el intervalo de ortogonalidad es $[0, \infty[$. Para graficar, use $[0, 8]$.
 - A la hora de graficar, quite la función `np.abs` en la línea
`polynomials[:,i]/np.abs(polynomials[0,i])`
de modo que se lea
`polynomials[:,i]/polynomials[0,i]`
(ver el ejemplo de Jupyter notebook citado).