

Concurso Ayudante de Segunda, área Materias que usan Herramientas Computacionales

A los 8 días del mes de septiembre de 2026, el jurado que entiende en el concurso para proveer 15 (quince) cargos de Ayudante de Segunda con dedicación Parcial, área Materias que usan Herramientas Computacionales en el Departamento de Matemática de la Facultad de Ciencias Exactas y Naturales (UBA), formado por Pablo Blanc, Mariela Cañizares Solis, Florencia Fernández Slezak, Tomás Fernández Vidal y Pablo Zadunaisky, decide asignar los siguientes puntajes máximos en los distintos ítems:

Antecedentes docentes	6
Antecedentes científicos	3
Antecedentes de extensión	5
Antecedentes profesionales	3
Prueba de oposición	58
Calificaciones, títulos, estudios y otros	25

Tema y modalidad de la prueba de oposición:

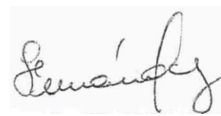
La prueba de oposición se realizará en forma presencial el día viernes 18 de septiembre de 2026 a las 14 horas, en los laboratorios 1109 y 1110 del Pabellón 0 + ∞ , y tendrá una duración de 60 minutos. El/La concursante deberá elaborar un programa utilizando el lenguaje de programación Python, siguiendo la consigna de uno de los ejercicios de la lista detallada a continuación. La lista consta de 7 ejercicios de programación con un nivel comparable a las materias Elementos de Cálculo Numérico, Álgebra Lineal Computacional y Matemática II para Biología. El ejercicio a desarrollar será anunciado por el jurado al inicio de la prueba de oposición. El programa debe ser claro e incluir los debidos comentarios para que un estudiante pueda comprender el código y el problema, tanto al comienzo y a lo largo del código como al final de la resolución. Todos los comentarios deben estar escritos dentro del mismo archivo .ipynb a entregar, que deberá nombrarse con el formato Apellido.Nombre.ipynb. Al finalizar la prueba, el archivo con el código debe ser enviado al correo electrónico concursoay2p26HC@gmail.com



Pablo Zadunaisky



Pablo Blanc



Florencia Fernández Slezak



Tomás Fernández Vidal



Mariela Cañizares Solis

Lista de ejercicios

Álgebra Lineal Computacional

1. Un sujeto en evidente estado de ebriedad oscila entre su casa y el bar, separados por n pasos. En cada instante de tiempo da un paso hacia adelante (acercándose a su casa), con probabilidad p o hacia atrás (acercándose nuevamente al bar), con probabilidad $1 - p$. Si llega a alguno de los dos extremos, se queda allí y no vuelve a moverse.
 - a) Sin hacer ninguna cuenta, mostrar que el proceso admite al menos dos estados límite linealmente independientes entre sí. Implementar un programa que reciba como input la distancia entre la casa y el bar (n) y la probabilidad p y devuelva la matriz de transición del proceso. Verificar que el resultado sea correcto corriéndolo para $n = 5$ y $p = 0,5$.
 - b) Para $n = 20$, tomar $p = 0,5$ y v_0 el vector que corresponde a ubicar al sujeto en cualquiera de los puntos intermedios del trayecto con igual probabilidad. Realizar una simulación del proceso hasta que se estabilice. ¿Cuál es el estado límite? ¿Cómo se interpreta?
 - c) Repetir la simulación tomando como vector inicial $v_0 = e_2$ (el segundo canónico). Interpretar el resultado.
 - d) Repetir las simulaciones con $p = 0,8$. ¿Qué se observa?
 - e) Explicar los resultados de todas las simulaciones a partir del análisis de los autovalores y autovectores de la matriz.

Matemática II (B)

2. Se quiere verificar numéricamente el orden de convergencia de los métodos de Euler y Taylor de orden 2. Para ello: resolver numéricamente el problema

$$\begin{cases} \dot{x}(t) = x(t), \\ x(t_0) = 1, \end{cases}$$

en el intervalo $[0, 1]$ con ambos métodos, tomando $h = 0,1, 0,05, 0,01, 0,005, 0,001, 0,0005$. Obtener la solución exacta y para cada h , calcular el error que se comete al aproximar $x(1)$: $E_N = |x(1) - x_N|$. Graficar $\log(E_N)$ en función de $\log(h)$. ¿Qué se espera ver? ¿El resultado es consistente con el esperado?

3.
 - a) Implementar un programa que reciba como input una función f , su derivada f' , un punto inicial x_0 y una tolerancia ϵ y aplique el método de Newton-Raphson para buscar una raíz de f a partir de x_0 . El programa debe finalizar cuando $|x_n - x_{n-1}| < \epsilon$.
 - b) Graficar $f(x) = (x + 1)e^x - 4$ en el intervalo $[0, 5]$ y proponer un intervalo de longitud 1, contenido en $[0, 5]$, que contenga una raíz de f .
 - c) Usando el programa del ítem a, verificar que el método de Newton-Raphson converge decrecientemente a la raíz determinada en el ítem anterior para los valores iniciales $x_0 = 1, 1,5, 2, 2,5, 3, 5$.

- d) ¿Qué sucede si $x_0 = 0,5$? (Mirar el valor de x_1 .)
 e) Repetir el análisis anterior para $g(x) = -f(x) = -(x+1)e^x + 4$.

Elementos de Cálculo Numérico

4. Sea S el subespacio de funciones continuas definidas de $\mathbb{R}$ en $\mathbb{R}$ generado por las funciones del conjunto $B = \{1, x, 2^x, 3^x\}$. Para $i = 0, 1, 2, 3$, sea $x_i = i$, y sea T un conjunto de datos del tipo $\{(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)\}$.

- a) Aproximar la siguiente tabla de datos en el sentido de cuadrados mínimos para el producto interno $\langle p, q \rangle = \sum_{i=0}^3 p(x_i)q(x_i)$

x	0	1	2	3
y	0.3	-0.2	7.3	23.3

con funciones del tipo: (a) $y = a2^x + b3^x$, (b) $y = a2^x + b3^x + c$.

- b) Graficar los resultados obtenidos junto con los valores de la tabla de datos.

5. Consideramos $f(x) = \frac{1}{1+25x^2}$ en el intervalo $[-1, 1]$:

- a) Interpolan $f(x)$ en $n+1$ puntos equiespaciados. Graficar simultáneamente la función con sus respectivos interpoladores para $n = 5, 10, 15$.
 b) Repetir el ítem anterior utilizando los ceros del polinomio de Tchebychev de grado $n+1$.
 c) Para ambos casos, calcular el error en norma infinito $\|f - P_n\|_\infty$. Explicar a qué se debe la diferencia en el comportamiento del error entre ambos tipos de nodos a medida que n aumenta y cómo se vinculan los valores obtenidos con los teoremas de acotación del error.

6. Se desea implementar una regla de cuadratura adaptativa, es decir, una cuadratura compuesta que utilice más subintervalos en la zona en que la aproximación obtenida sea peor. Para ello, notamos $S(a, b)$ a la regla de Simpson en el intervalo $[a, b]$. Si notamos $c = \frac{a+b}{2}$ se tiene que:

$$|S(a, b) - S(a, c) - S(c, b)|/15 \sim E(a, c, b)$$

donde $E(a, c, b)$ es el error cometido al aplicar la regla compuesta: $S(a, c) + S(c, b)$. Implementar un programa que reciba como input una función f , un intervalo $[a, b]$ y una tolerancia ϵ y calcule las cuadraturas: $q = S(a, b)$, $q_1 = S(a, c)$ y $q_2 = S(c, b)$. Si $|q - q_1 - q_2| < 15\epsilon$ se devuelve el valor $q_1 + q_2$. En caso contrario, se aplica el mismo criterio para integrar f en los intervalos $[a, c]$ y $[c, b]$, con una tolerancia $\epsilon/2$.

Probar el programa calculando $\int_0^1 xe^{-x^2} dx = \frac{1}{2}(1 - e^{-1})$. Comparar los resultados (y los tiempos de ejecución) con los obtenidos por la regla de Simpson compuesta.

7. Implementar dos programas que calculen la descomposición QR de una matriz:

- a) Aplicando de manera directa el método de ortonormalización de Gramm-Schmidt.
 b) Utilizando transformaciones de Householder.

Generar algunas matrices aleatorias y comparar las descomposiciones arrojadas por estos programas con las dadas por el comando `np.linalg.qr`. ¿Qué se observa?