Considere la siguiente ecuación diferencial:

$$y''(x) = (y(x) - 1)(1 + (y'(x))^2)^{3/2}, \text{ para } x \in]-1, 1[,$$
$$y(\pm 1) = 0$$

Determine:

1. Construya un algoritmo que obtenga una aproximación numérica de $y(x)$ considerando una discretización de N puntos del intervalo $[-1, 1]$.

- **[20 puntos] [Se revisará offline]** Implemente *'solve_2nd_order_ODE'*.

2. Obtenga $\int_{-1}^{1} y(s)\, ds$ considerando una discretización de N puntos.

- **[4 puntos][Se revisará offline]** Implemente *'compute_integral' utilizando el método trapecio.*
- **[2 puntos]** Utilice *'compute_integral'* para obtener la aproximación con $N = 100$, su resultado debe incluir por lo menos 5 decimales.

0.54

✔

Habian dos opciones posibles para la solución de la ODE, usar diferencias finitas o el metodo del disparo. Las posibles respuestas eran 0.54573 ó 0.54566, por lo que responder cualquiera de ellas con un error absoluto menor a 1e-5 se consideraba correcto. El 0 de la retroalimentación viene solo como placeholder, indicando a la plataforma que la respuesta era una sola. La calificación depende de operaciones logicas que concideraban como correcta cualquiera de los pares antes mencionados.

Una posible respuesta correcta sería: 0

3. El valor máximo de $y(x)$ en $x \in]-1, 1[$ considerando una discretización de N puntos.

- **[8 puntos] [Se revisará offline]** Implemente *'find_max'*.
- **[2 puntos + 2 puntos]** Utilice *'find_max'* para obtener la aproximación con $N = 100$.

max_y = 0.38, max_x = 0.01

✔

Al igual que en la sección anterior, habían dos respuestas posibles:

- max_y = 0.38909 , max_x = -0.01010
- max_y = 0.38903 , max_x = 0.01010

De la misma forma, responder con un error absoluto menor a 0.00001 era considerado como correcto. El 0 de la retroalimentación viene solo como placeholder, indicando a la plataforma que la respuesta era una sola. La calificación depende de operaciones logicas que concideraban como correcta cualquiera de los pares antes mencionados.

Una posible respuesta correcta sería: 0, 0

4. Construya un algoritmo que determine $\widehat{x}$ sobre la grilla discreta de N puntos tal que la pendiente de $y(\widehat{x})$ aproximadamente m, para aceptar la aproximación se debe cumplir que $|y'(\widehat{x}) - m| < \delta$. Por simplicidad, se informa que la pendiente se validará con diferencias finitas de segundo orden sobre la grilla de puntos equiespaciados. Notar que esto no restringe que el problema se resuelva con diferencias finitas. En caso de que no se encuentre una aproximación válida de $\widehat{x}$ para m, se debe retornar *np.nan*.

- **[8 puntos][Se revisará offline]** Implemente *'find_widehat_x'*.
- **[2 puntos]** Utilice *'find_widehat_x'* para obtener la aproximación con $m = 0.0044$, $N = 100$, $delta = 1$.

-0.9

✖

Independiente del metodo utilizado, la respuesta truncada a 5 decimales es -0.17171, por lo que cualquier entrada con un error absoluto menor o igual a 0.00001 sería correcto.

Una posible respuesta correcta sería: -0.17171

- **[2 puntos]** Utilice *'find_widehat_x'* para obtener la aproximación con $m = 0.0044$, $N = 100$, $delta = 10^{-4}$.

2

✔

De igual manera que en la pregunta anterior, la respuesta no depende del método utilizado, pues en ambos casos es np.nan. De acuerdo a los anuncios realizados durante el certamen, esta respuesta debía ser ingresada como 2.

Una posible respuesta correcta sería: 2

Considere la siguiente ecuación matricial:

$$X + A^T X^{-1} A = Q$$

donde las matrices A, X, y Q pertenecen a $\mathbb{R}^{n \times n}$, A es no singular, ^{T} corresponde al operador transpuesta y $^{-1}$ corresponde al operador inversa. Recordar que **no está permitido calcular la inversa de una matriz de forma explícita**, es decir, no debe obtener X^{-1} ni ninguna inversa, sino **resolver el sistema de ecuaciones lineales asociado**.

Considere la siguiente manipulación algebraica para la construcción de una iteración de punto fijo en alta dimensión,

$$\begin{aligned} X + A^T X^{-1} A &= Q, \\ X A^{-1} + A^T X^{-1} &= Q A^{-1}, \\ X A^{-1} X + A^T &= Q A^{-1} X, \end{aligned}$$

lo que nos da la siguiente iteración de punto fijo en alta dimensión,

$$X^{(m+1)} A^{-1} X^{(m)} + A^T = Q A^{-1} X^{(m+1)}.$$

1. Construya un algoritmo que obtenga $X^{(m+1)}$ y su residuo relativo utilizando norma 2 dado A, Q y $X^{(m)}$. Considere que el sistema de ecuaciones lineales resultante es no singular y que A es no singular tampoco. El solver debe entregar la solución considerando en residuo relativo menor o igual a 10^{-10} considerando la norma 2.

- **[40 puntos][Se revisará offline]** Implemente *'obtain_next_X'*. Considere el siguiente caso particular como **test case** : $A = I$, $X^{(m)} = I$, y $Q = 2\,I$, donde I es la matriz identidad de orden n. La solución de este caso particular es $X^{(m+1)} = I$. Este **test case** es válido para cualquier valor de n finito y mayor o igual a 1.

- **[20 puntos]** Obtenga la solución numérica para el siguiente caso:

```
A  = np.array([[10.5488135,   0.71518937], [0.60276338, 10.54488318]])
Q  = np.array([[2.37096596, 0.6638646 ], [0.6638646,  3.21243829]])
Xm = np.array([[1.0, 0.0 ], [0.0,  1.0]])
n = 2
```

considerando que el output es

$$\text{Xout} = \begin{pmatrix} x_{0,0} & x_{0,1} \\ x_{1,0} & x_{1,1} \end{pmatrix},$$

$x_{0,0}$ = 90.9|, $x_{0,1}$ = -21.1|, $x_{1,0}$ = -18.1|, $x_{1,1}$ = 56.3|, y $\log_{10}(\text{rel\_residual})$=-14.2.

Se deben incluir por lo menos 5 decimales para cada una de las 4 componentes de la matriz Xout como para el logaritmo en base 10 del residuo relativo.

Hint: You should not build the associated large linear system of equations, so it may be useful to recall professor Aleksey Nikolaevich Krylov.

Hint2: Recall that if you have a linear system of equations $A\mathbf{x} = \mathbf{b}$ you compute it relative residual as $\|\mathbf{b} - A\widehat{\mathbf{x}}\|/\|\mathbf{b}\|$, where $\widehat{\mathbf{x}}$ is the numerical approximation found. So, in case you don't have explicit access to the corresponding matrix A, you can still use any routine you may have available to compute the matrix-vector product $A\widehat{\mathbf{x}}$, it may be aFUN function!

☑

El output de la función que se pedía debía ser lo siguiente:

Xout:

[[90.91574114 -21.15918122]

[-18.15328002 56.3580065]]

np.log10(rel_residual): -15.446616666304829

Una posible respuesta correcta sería: 0, 0, 0, 0, 0