

PVCM\_Test 4

Statut	Terminée
Commencé	lundi 16 juin 2025, 10:44
Terminé	lundi 16 juin 2025, 11:53
Durée	1 heure 9 min

Question 1

Correct

Noté sur 2,00

Marquer la question

Écrire la fonction qui retour les valeurs des indices impairs de la diagonale d'une matrice. / Write a function which returns the values of odd indices of the diagonal of a matrix.

A =  $\begin{vmatrix} 6 & 3 & 2 \\ 4 & 5 & 8 \\ 0 & 4 & 7 \end{vmatrix}$

onetwo(A) returns np.array([6, 7])

Fonction en 2 lignes / Function of 2 lines

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def onetwo(A):
    return np.diag(A)[::2]
```

Test	Résultat attendu	Résultat obtenu	
res = onetwo(np.array([[1,2],[1,2]])) print(np.all(res == np.array([1])))	True	True	

Tous les tests ont été réussis !

Question 2

Incorrect

Noté sur 2,00

Marquer la question

Écrire la fonction qui donne la position (i,j) de la première plus petite valeur positive de la matrice donnée en argument. / Write a function which return the position (i,j) of the first smallest positive value of the matrix given in argument.

A =  $\begin{vmatrix} 6 & 3 & -2 \\ 4 & -5 & 8 \\ 0 & 4 & 0 \end{vmatrix}$

posmin(A) returns (2,0)

Fonction en 3 lignes max / Function in 3 lines max.

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def posmin(A):
    return np.argmin(A)
```

Test	Résultat attendu	Résultat obtenu	
print(posmin(np.arange(9).reshape(3,3)) == (0,0))	True	[ True True]	

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

Montrer les différences

Question 3

Non répondue

Noté sur 4,00

Marquer la question

Écrire une fonction qui retourne la matrice A exprimée dans la base de ses vecteurs propres. / Write a function which returns the matrix A in its eigenvector base.

L'utilisation de linalg est autorisée. / You can use linalg.

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def base_eig(A):
```

Question 4

Correct

Noté sur 6,00

Marquer la question

Décomposer A en A = B B.T avec B une matrice triangulaire inférieure. La fonction retourne B. / Factorize A to B @ B.T where B is a lower triangular matrix. The function returns B.

Une seule boucle for est autorisée. / Only one loop for authorized.

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def Choleski(A):
    res = np.zeros(A.shape)
    for i in range(len(A)):
        res[i, i] = np.sqrt(A[i, i] - np.sum(np.square(res[i, :i])))
        res[i+1:, i] = (A[i, i+1:] - res[i, :i] @ res.T[:i, i+1:]) / res[i, i]

    return res
```

Test	Résultat attendu	Résultat obtenu	
A = np.random.randint(10, size=(5,5)) A = A + A.T A = A + np.diag(A.sum(axis=0)) B = np.zeros(A.shape) for i in range(len(A)): B[i,i] = np.sqrt(A[i,i] - np.sum(np.square(B[i, :i]))) B[i+1:, i] = (A[i, i+1:] - B[i, :i] @ B.T[:i, i+1:]) / B[i,i] B2 = Choleski(A) print(np.max(np.abs(B - B2)) < 1E-5)	True	True	

Tous les tests ont été réussis !

Question 5

Non répondue

Noté sur 6,00

Marquer la question

Résoudre le système linéaire A x = b par la méthode du gradient. On s'arrête lorsque le résidu < 1E-3. Le pas mu = 1E-2. La sortie de la fonction est un tableau numpy en 2D avec pour chaque ligne la position de l'itération (inclus la position initiale x0).

Solve A x = b with the gradient descent. Stop descent when the residual error < 1E-3. The step size mu = 1E-2. The output of the function is an 2D array with position for each iteration including the first position x0 on the first line.

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def gradient(A,b, x0, mu = 1E-2, error=1E-3):
```

Test	Résultat attendu	Résultat obtenu
A = np.array([[3,1],[1,3]]) b = np.array([4,4]) res = gradient(A,b, np.zeros(2)) print(np.sum(np.square(A @ res[-1] - b)) < 1E-6)	True	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 32, in <module> print(np.sum(np.square(A @ res[-1] - b)) < 1E-6) numpy.core._exceptions._UFuncNoLoopError: ufunc 'matmul' did not c

Le test a été interrompu à cause d'une erreur.

Montrer les différences