

numpy_td.py

```
#####  
#####Numpy: TD-TP:16#####  
# #####
```

Exercice 1

Pour dessiner la fonction $f(x)$ en utilisant la bibliothèque numpy, on procède comme suit:

1. génère une subdivision de l'intervalle $[a,b]$ x en utilisant `np.linspace()`,
2. déterminer le vecteur image $y=f(x)$
3. trace la fonction en utilisant `plt.plot()`.

Les fonctions `plt.xlabel()`, `plt.ylabel()` et `plt.title()` sont utilisées pour ajouter des étiquettes et un titre au graphique.

4. `plt.show()` affiche le graphique. Vous pouvez ajuster la plage de valeurs x en modifiant les arguments de `np.linspace()`.

Question: Dessiner $f(x)=\exp(x)+\tan(x)/\log(x^2+1)$ sur $[-5,5]$

Exercice 2

Question: Dessiner $f(x)=\exp(x)$, $g(x)=$ et $h(x)$ sur le même graphe sur $[0,1]$

Exercice 3

1. Créez un tableau NumPy contenant les nombres de 1 à 10.
2. Créez un tableau NumPy de 5x5 rempli de zéros.
3. Créez un tableau NumPy de 5x5 rempli de uns.
4. Créez un tableau NumPy de 5x5 rempli de nombres aléatoires entre 0 et 1.
5. Créez un tableau NumPy de 10 éléments, puis divisez-le en deux tableaux de 5 éléments.
6. Créez un tableau NumPy de 10x10 contenant les nombres de 1 à 100, puis extrayez la première colonne.
7. Créez deux tableaux NumPy de 5 éléments chacun, puis calculez leur somme et leur produit.
8. Créez un tableau NumPy contenant les nombres pairs de 2 à 20, puis calculez sa moyenne.
9. Créez un tableau NumPy contenant les nombres impairs de 1 à 25, puis calculez sa somme cumulée.
10. Créez un tableau NumPy contenant les notes de 5 élèves à un examen, puis calculez la moyenne et l'écart-type.

Exercice 4 (fonction transposée)

En utilisant numpy, la transpose d'une matrice `m` donne est déterminée par:
`tm=m.T` ou `tm=np.transpose(A)`

Ecrire une fonction, nommée `transpose`, qui prend en paramètre une matrice et qui renvoie la matrice transposée sans utiliser la méthode `.T`

Exercice 5

Transformation matricielle:

1. Calculez **la** transposée **de la** matrice $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$.
2. Multipliez **la** matrice $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ par **la** matrice $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$.
3. Calculez le **déterminant de la** matrice $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.
4. Calculez **l'inverse de la** matrice $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.
5. Calculez **la** matrice diagonale **D** et **la** matrice triangulaire supérieure **U** telles **que** $A = DU$, où $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$.

Exercice 6

Ecrire une fonction `mat_vect` **qui** prend **en** paramètre une matrice et **qui** renvoie le vecteur **de** dimension 1 contenant tous les éléments **de la** matrice.

Exemple :

```
'''
>>> m=np.array([[1,2,3],[4,5,6],[7,8,9]])
>>> mat_vect(m) renvoie np.array([1,2,3,4,5,6,7,8,9])
'''
```

Exercice 7

La somme **de** Riemann **est** une méthode pour approximer **l'intégrale d'**une fonction **continue** $f(x)$ sur un intervalle donné $[a, b]$. **La** méthode consiste à diviser **l'intervalle** $[a, b]$ **en n** sous-intervalles égaux et à utiliser **la** valeur **de** $f(x)$ sur chaque sous-intervalle pour calculer une somme pondérée. **La** somme **de** Riemann **est** alors donnée par **la** limite **de** cette somme pondérée lorsque **n** tend **vers l'infini**.

Pour calculer **la** somme **de** Riemann **de la** fonction $f(x) = \exp(x)$ sur un intervalle $[a, b]$, voici les étapes à suivre :

- a- Divisez **l'intervalle** $[a, b]$ **en n** sous-intervalles égaux **de** largeur $\Delta x = (b-a)/n$.
- b- Calculez les valeurs **de** $f(x)$ pour chaque point **de la** grille $x_i = a + i \cdot \Delta x$, où $i = 0, 1, \dots, n$.
- c- Calculez **la** somme pondérée **de** ces valeurs **en** multipliant chaque $f(x_i)$ par Δx , et **en** sommant le tout : $S_n = \Delta x * (f(x_0) + f(x_1) + \dots + f(x_{n-1}))$.
- d- Prenez **la** limite **de** cette somme lorsque **n** tend **vers l'infini** pour obtenir **l'intégrale de** $f(x)$ sur $[a, b]$.

- 1- **En** utilisant Python, calculer **la** somme **de** Riemann **de la** fonction $f(x) = \exp(x)$ sur **l'intervalle** $[0, 1]$ avec $n = 100$:
- 2- Ecrire une fonction `reiman_int`, **qui** accepte une fonction f , a et b les bornes **de l** intervalle, et n le nombre **de** points et permet **de** calculer et retourner **la** somme **de** Riemann **de la** fonction f sur **l'intervalle** $[a, b]$.

Application:

- i. Calculez **l'intégrale définie de** $f(x) = x^2 + 2x + 1$ sur **l'intervalle** $[0, 2]$.
- ii. Calculez **l'intégrale définie de** $f(x) = \sin(x)$ sur **l'intervalle** $[0, \pi]$.