

TD 18 -

Calcul approché de dérivées et primitives.

Informatique
MPSI-PCSI - Lycée Thiers

Exercice 1 : Tracé de $\arctan$ et de sa dérivée

Enoncé

Réponse

Exercice 2 - Tracé du graphe de $\ln$ par primitivation numérique

Enoncé

Corrigé

Exercice 3 - Tracé de $\exp$ par primitivation numérique

Enoncé

Exercice 1 : Tracé de $\arctan$ et de sa dérivée

Exercice 1.

1. Tracer le graphe de la fonction $\tan$ sur l'intervalle $\left[-\frac{\pi}{2} + 0, 1; \frac{\pi}{2} - 0.1\right]$.
2. La fonction *tangente* réalise une bijection de $\left]-\frac{\pi}{2}; \frac{\pi}{2}\right[$ sur $\mathbb{R}$. A l'aide des tableaux obtenus pour le tracé précédent, tracer sur un nouveau graphique, le graphe de sa bijection réciproque $\arctan$.
3. Sur un nouveau graphique, tracer le graphe de la dérivée de $\arctan$. On utilisera pour cela un calcul approché, le plus précis possible, sur les tableaux constitués à la question 1.

Exercice 1 : corrigé

- 1) Tracé :

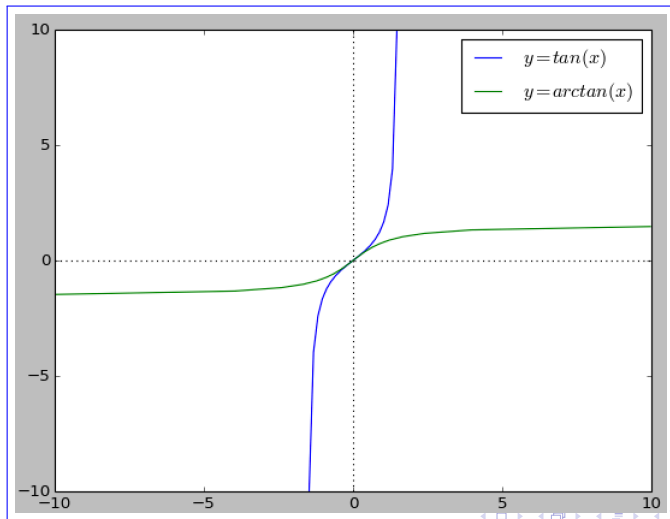
```
import numpy as np
import matplotlib.pyplot as plt
A = np.pi/2-0.1
N=1000

X = np.linspace(-A,A,N+1)
Y = np.tan(X)

plt.figure(1)
plt.clf()
plt.plot(X,Y) # tan
plt.plot(Y,X) # arctan
plt.plot([0,0],[ -10,10], 'k:', [-10,10],[0,0], 'k:')
plt.legend((' $y=\tan(x)$ ', ' $y=\arctan(x)$ ',))
```

Exercice 1 : corrigé

- 1) Tracé :



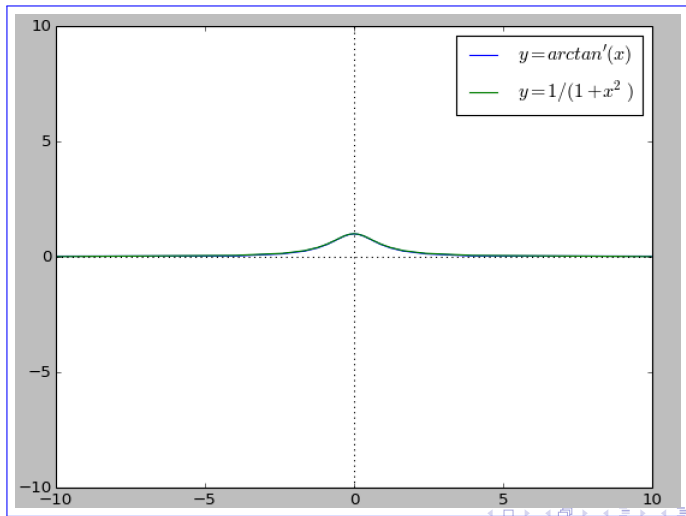
Exercice 1 : corrigé

2) Tracé de $\arctan'$ par dérivation numérique.

```
A = Y
O = X
dF = np.empty(N+1)
dF[0] = (O[1]-O[0]) / (A[1]-A[0])
dF[N] = (O[N]-O[N-1]) / (A[N]-A[N-1])
for k in range(1,N):
    dF[k] = (O[k+1]-O[k-1]) / (A[k+1]-A[k-1])
plt.figure(2)
plt.clf()
plt.axis('equal')
plt.plot(A,dF)
plt.plot(A, 1/(1+A**2))
plt.plot([0,0],[-10,10], 'k:', [-10,10],[0,0], 'k:')
plt.legend("$y=\arctan'(x)$", '$y=1/(1+x^2)$',)
```

Exercice 1 : corrigé

- 2) Tracé :



Exercice 2 - Enoncé

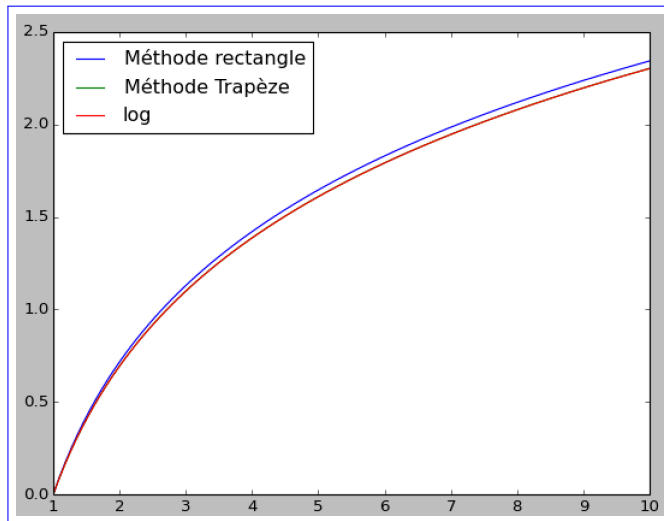
Exercice 2. On rappelle que la fonction $\ln$ est la primitive de $x \mapsto \frac{1}{x}$ sur $\mathbb{R}_+^*$ valant 0 en 1.

1. Sans utiliser la fonction `log` des modules `numpy` ou `math` de python, utiliser la méthode des rectangles pour obtenir des valeurs approchées prises par la fonction $\ln$ sur une subdivision régulière de 1001 points de l'intervalle $[1, 10]$, puis tracer son graphe.
2. Faire de même avec la méthode des trapèzes.
3. Sur le même graphique tracer aussi le graphe de $\ln$ obtenu grâce à la fonction `log` de `numpy`.

Exercice 2 - Corrigé

```
A = 10
N = 100
X = np.linspace(1,A,N+1)
f = lambda x: 1/x
dx = (A-1)/N
L = np.empty(N+1)      # 1) Méthode rectangle
L[0]=0
Lt = np.empty(N+1)     # 2) Méthode Trapèzes
Lt[0]=0
for k in range(N):
    L[k+1] = L[k] + dx * f(X[k])          # 1)
    Lt[k+1] = Lt[k] + dx * (f(X[k])+f(X[k+1]))/2  # 2)
plt.figure(2)
plt.clf()
plt.plot(X,L)
plt.plot(X,Lt)
plt.plot(X,np.log(X))      # 3) log
plt.legend(('Méthode rectangle', 'Méthode Trapèze', 'log'),
'upper left')
```

Exercice 2 - Corrigé



Exercice 3 - Enoncé

Exercice 3. On rappelle que la fonction $\exp$ est sa propre primitive sur $\mathbb{R}$ valant 1 en 0.

1. Sans utiliser la fonction `exp` des modules `numpy` ou `math` de python, utiliser la méthode des rectangles pour obtenir des valeurs approchées prises par la fonction $\exp$ sur une subdivision régulière de 1001 points de l'intervalle $[0, 2]$, puis tracer son graphe.
2. Faire de même avec la méthode des trapèzes.
3. Sur le même graphique tracer aussi le graphe de $\exp$ obtenu grâce à la fonction `exp` de `numpy`.
4. Sauriez-vous tracer le graphe de $\exp$ au-dessus $[-2, 2]$ sans utiliser la fonction `exp` d'un module de python ?

Exercice 3 - Remarque

1) On utilise $\exp' = \exp$ et $\exp(0) = 1$.

Primitivation par la méthode des rectangles :

$$\exp(x + dx) = \exp(x) + dx \times \exp(x)$$

```
plt.figure(3)
A = 2
N = 1000
X = np.linspace(0,2,N+1)
dx = A/N
Y = np.empty(N+1)
Y[0]=1
for k in range(N):
    Y[k+1] = Y[k]+dx*Y[k]
```

Exercice 3 - Remarque

1) On utilise $\exp' = \exp$ et $\exp(0) = 1$.

Primitivation par la méthode des trapèzes :

$$\exp(x + dx) = \exp(x) + dx \times (\exp(x) + \exp(x + dx)) / 2$$
$$\implies \exp(x + dx) = \frac{2 + dx}{2 - dx} \exp(x)$$

```
Yt = np.empty(N+1)
Yt[0]=1
for k in range(N):
    Yt[k+1] = Yt[k] * (2+dx)/(2-dx)
plt.plot(X,Y,X,Yt,X,np.exp(X))
plt.legend(('M. Rectangle', 'M. Trapèze', 'y = exp(x)'),loc='upper
left')
```

Exercice 3 - Corrigé

