Cours_Calcul_Integral_2020

March 17, 2020

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

```
In [2]: %matplotlib inline
```

0.1 Méthode des rectangles

```
In [3]: def rectangleGauche(f, a, b, n):
            s = 0
            pas = (b - a)/n
            x = a
            for k in range(n):
                s = s + f(x) * pas
                x = x + pas
            return s

        def rectangleGauche2(f, a, b, n):
            s = 0
            pas = (b - a) / n
            for k in range(n):
                s = s + f(a + k * pas) * pas
            return s

        def rectangleGaucheDessin(f, a, b, n):
            s = 0
            pas = (b - a)/n
            x = a
            for k in range(n):
                s = s + f(x) * pas
                plt.fill([x, x + pas, x + pas, x, x], [0] * 2 + [f(x)] * 2 + [0], hatch='/',edg
                x = x + pas
            lesx = np.linspace(a, b, 1000)
            fvect = np.vectorize(f)
            ax = plt.gca()
            ax.spines['top'].set_color('none')
            ax.spines['right'].set_color('none')
            ax.spines['bottom'].set_position(('data',0))
```

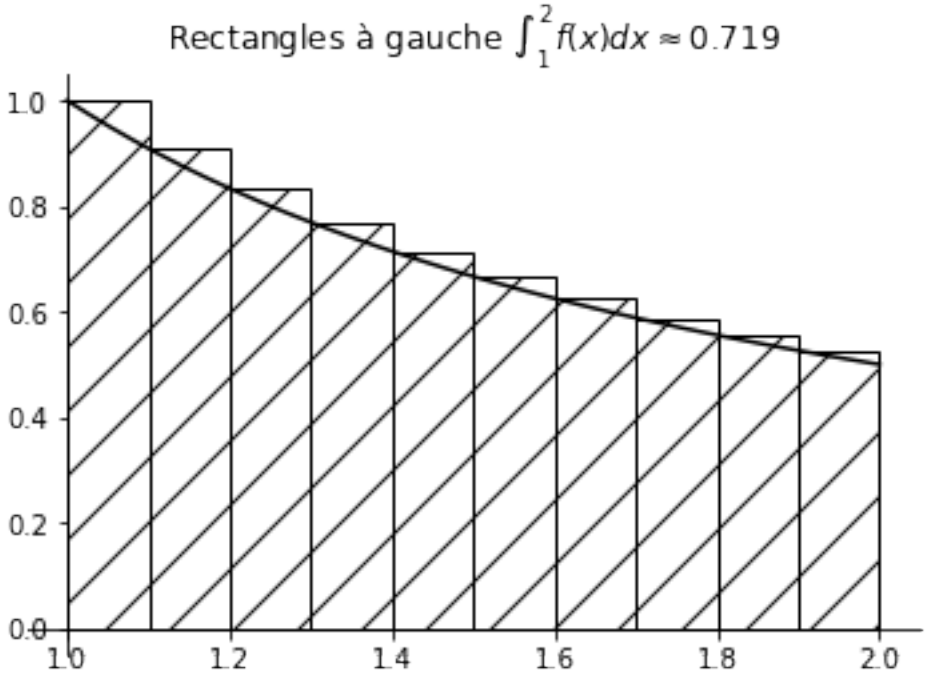
```
            ax.spines['left'].set_position(('data',a))
            plt.plot(lesx, fvect(lesx), color='black')
            plt.title(r"Rectangles à gauche $\int_{%d}^{%d}f(x)dx\approx$%1.3f"%(a,b,s))
            plt.savefig('methodeRectangleGauche-{}-{}-{}-{}subdivisions.eps'.format(f.__name__
            plt.show()
            return s
```

In [4]:
```
def rectangleDroite(f, a, b, n):
    s = 0
    pas = (b - a)/n
    x = a + pas
    for k in range(n):
        s = s + f(x) * pas
        x = x + pas
    return s

def rectangleDroite2(f, a, b, n):
    s = 0
    pas = (b - a) / n
    for k in range(1, n + 1):
        s = s + f(a + k * pas) * pas
    return s

def rectangleDroiteDessin(f, a, b, n):
    s = 0
    pas = (b - a)/n
    x = a
    for k in range(n):
        s = s + f(x + pas) * pas
        plt.fill([x, x + pas, x + pas, x, x], [0] * 2 + [f(x+pas)] * 2 + [0], hatch='/
        x = x + pas
    lesx = np.linspace(a, b, 1000)
    fvect = np.vectorize(f)
    ax = plt.gca()
    ax.spines['top'].set_color('none')
    ax.spines['right'].set_color('none')
    ax.spines['bottom'].set_position(('data',0))
    ax.spines['left'].set_position(('data',a))
    plt.plot(lesx, fvect(lesx), color='black')
    plt.title(r"Rectangles à droite $\int_{%d}^{%d}f(x)dx\approx$%1.3f"%(a,b,s))
    plt.savefig('methodeRectangleDroite-{}-{}-{}-{}subdivisions.eps'.format(f.__name__
    plt.show()
    return s
```

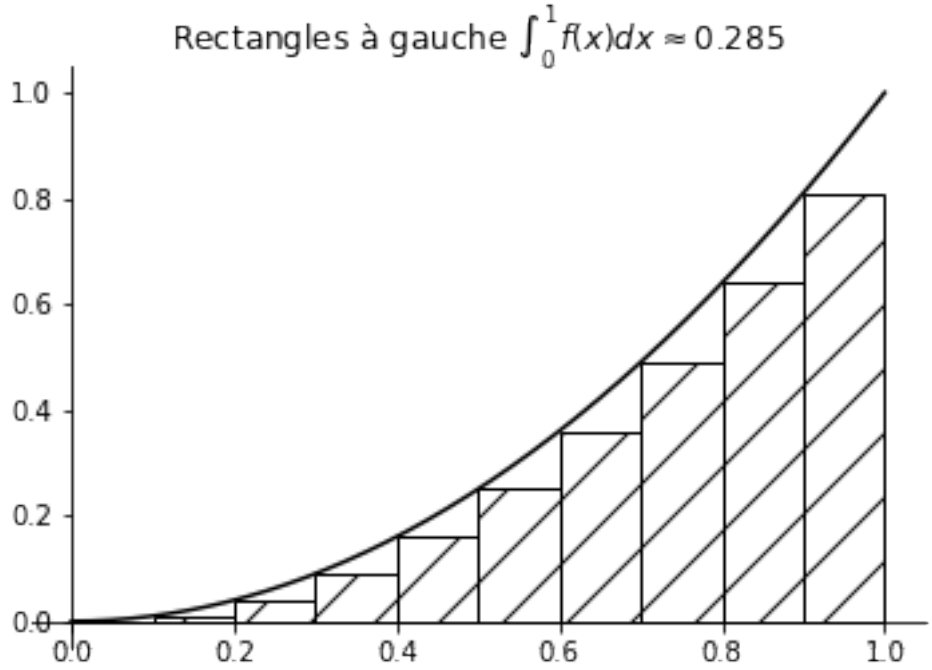
In [6]:
```
def inverse(x):
    return 1 / x
```

In [7]:
```
inverse.__name__
```

```
Out[7]: 'inverse'

In [8]: rectangleGauche(inverse, 1, 2, 10)

Out[8]: 0.7187714031754279

In [9]: rectangleGauche2(inverse, 1, 2, 10)

Out[9]: 0.718771403175428

In [10]: rectangleGaucheDessin(inverse,1,2,10)
```

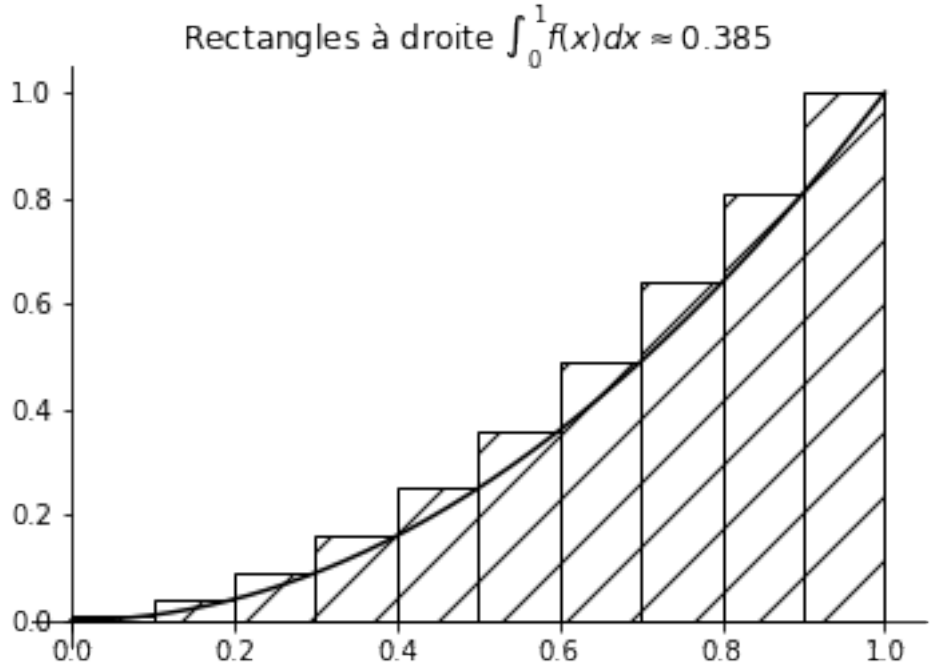



```
Out[10]: 0.7187714031754279

In [68]: def carre(x):
             return x ** 2

In [104]: rectangleGaucheDessin(carre,0,1,10)
```



Out[104]: 0.2849999999999999

In [105]: rectangleDroiteDessin(carre, 0, 1, 10)



```
Out[105]: 0.3849999999999999
```