

Dans ce TD, nous allons nous intéresser à différentes approches pour calculer l'aire de surfaces du plan. Nous aurons besoin des modules suivants

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

## I Calcul d'intégrale



Dans ce TD nous allons utiliser numpy. Les fonctions numpy agissent sur les vecteurs et les liste terme à terme pour les opérations usuelles ainsi que pour les fonctions de ce module. Par exemple

```
1 a1 = np.array([1,2,3]) # un vecteur numpy
2 a2 = np.array([4,5,6]) #
3 a1 + a2 # crée un vecteur contenant 5, 7, 9
4 a1 * a2 # crée un vecteur contenant 4, 10, 18
5 np.sqrt(a1) # crée un vecteur contenant 1, racine de 2 et racine de 3
```

### Exercice 1

On souhaite pour commencer tracer le cercle unité d'équation  $x^2 + y^2 = 1$ . On remarque que le demi-cercle supérieur est d'équation  $y = \sqrt{1 - x^2}$  et le demi cercle inférieur d'équation  $y = -\sqrt{1 - x^2}$ . Nous devons ainsi tracer deux courbes sur le même graphique.

En utilisant la commande `np.linspace` et les fonctions numpy tracer les deux demis-cercles précédents, en utilisant une seule couleur. On pourra consulter le TP 4.

On pourra au besoin utiliser l'aide python :

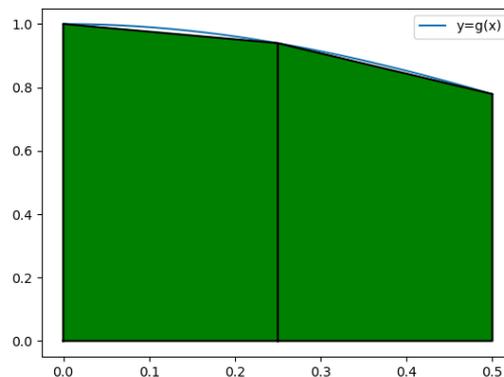
```
1 help(np.linspace)
```

### Exercice 2

On souhaite évaluer l'aire  $\mathcal{A}$  du cercle de rayon 1 en calculant l'intégrale  $\int_0^1 \sqrt{1 - x^2} dx$ .

1. Donner le lien entre l'aire  $\mathcal{A}$  cherchée et l'intégrale considérée.
2. Créer une fonction  $f$  en python, qui sera la fonction que l'on souhaite intégrer.  
Rappel : la valeur de  $f(x)$  est la valeur **retournée** par notre fonction python.

3.  La méthode des trapèzes pour estimer numériquement l'intégrale  $\int_a^b g(t)dt$  consiste à découper l'intervalle d'intégration en  $n$  segments égaux et approximer chaque morceau de la courbe de  $g$  par une droite, comme illustré sur la figure suivante où on a pris  $n = 2$



- (a) Donner en fonction de  $a, b, n$  les coordonnées de chaque point en commun entre la courbes et les trapèzes successifs. Par exemple le premier point est de coordonnées  $(a, g(a))$  et le dernier  $(b, g(b))$ . Donner les  $n - 1$  autres points.

- (b) Créer une fonction `trapeze(g, a, b, n)` qui calcule l'intégrale  $\int_a^b g(t)dt$  par la méthode des trapèzes en utilisant exactement  $n$  trapèzes.



L'aire d'un trapèze de hauteur  $h$  et dont les deux bases sont de longueurs  $b$  et  $B$  vaut  $h \times \frac{b+B}{2}$

Pour tester votre fonction, on prendra  $n = 10^4$ , la fonction  $f$  créée précédemment et des bornes bien choisies. Comparer à la valeur attendue en faisant varier la valeur de  $n$ .

4. Grâce à l'aide python, interpréter le résultat de

```
1 import scipy.integrate as sci
2 sci.quad(f, 0, 1)
```

Que dire de la précision de la méthode des trapèzes lorsque  $n = 1000$  ?  $n = 10^5$  ?

### Exercice 3 (Une application en probabilité)

En mathématique (en probabilités plus exactement), on définit la fonction `erf` par

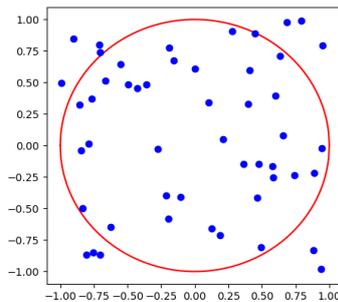
$$\forall x \in \mathbb{R} \text{ erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

- Définir en python la fonction `erf` correspondante. On pourra utiliser la méthode des trapèzes ou la fonction `quad` précédente.
- Tracer la courbe représentative de `erf` pour  $x \in [-3, 3]$ , en utilisant 250 points.
- Tracer sur le même graphique la courbe de la fonction `scs.erf` disponible après `limport` suivant :

```
1 import scipy.special as scs
```

## II Approche aléatoire

Une autre approche pour estimer l'aire d'une surface, est représentée sur la figure suivante. Le principe est de générer une



grand nombre de points aléatoires et de dénombrer les points à l'intérieur de la surface.

### Exercice 4 (Approche théorique)

On se place dans un carré de côté 2, délimité par  $x = \pm 1$ ,  $y = \pm 1$ .

- On note  $\mathcal{A}$  l'aire du disque unité que l'on cherche à estimer ici, et  $\mathcal{C}$  l'aire du carré considéré. Que vaut  $\mathcal{C}$  ?
- Imaginons que l'on choisisse  $N$  points  $A_1, \dots, A_N$  points dans le carré et que l'on obtienne  $n \leq N$  points à l'intérieur du cercle. Quelle quantité est approximée par  $\frac{n}{N}$  ?
- Pour un point fixé  $A_i = (x_i, y_i)$ , comment vérifier si  $A_i$  est à l'intérieur du disque ?
- Déduire des questions précédentes une approximation de  $\mathcal{A}$  en utilisant les notations précédentes.

### Exercice 5

Nous allons passer à la pratique. Pour cela nous allons avoir besoin d'un module gérant l'aléatoire.

```
1 import random
```

La fonction `random.random` retourne un flottant choisi aléatoirement dans  $[0, 1]$ .

1. Écrire une fonction `nombre_alea()` qui ne prend pas d'argument et retourne un flottant aléatoire dans  $[-1, 1]$ .
2. Écrire une fonction `nombre_points_disque(N)` qui prend en argument un entier  $N$ , tire aléatoirement  $N$  points dont les coordonnées sont dans  $[-1, 1]$  et retourne l'entier  $n$  évoqué à l'exercice précédent : le nombre de points appartenant au disque.
3. En utilisant la fonction précédente, obtenir des approximations de  $\pi$  pour  $N = 10^3, 10^4, 10^5, 10^6$ . Comparer à la méthode des trapèzes.
4. Bonus : créer une figure semblable à celle de l'illustration.

### III Pour aller plus loin

#### Exercice 6

On souhaite illustrer la méthode des trapèzes. Les données sont :

- une fonction python  $f$ ,
- deux flottants  $a, b$  tels que  $a < b$ .
- un entier  $n > 0$ .

On veut afficher sur un même graphique la courbe représentative de  $f$  sur  $[a, b]$  ainsi que  $n$  trapèzes dont les aires approximent l'intégrale de  $f$ . On utilisera

```
1 plt.fill(X, Y)
```

Pour remplir un polygone dont les sommets sont données par les listes d'abscisses  $X$  et d'ordonnées  $Y$ .

#### Exercice 7

On souhaite étudier un nuage de points donné (par exemple nos points aléatoires).

Ces points seront donnés dans une liste `points` qui contient des listes de longueur 2.

1. Comment trouver le point d'abscisse minimale ?
2. Sur une feuille, placer 8 points au hasard et tracer le plus petit polygone convexe (pas d'angles rentrant, on ne veut pas de étoile) qui englobe ces points. On appelle ce polygone l'enveloppe convexe.
3. Comment déterminer<sup>1</sup> l'enveloppe convexe de `points` ?

---

1. Difficile !