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

```python
import numpy as np
import matplotlib.pyplot as plt
```

# I Calcul d'intégrale

**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 $y=-\sqrt{1-x^2}$.
En utilisant la commande `np.linspace` et les fonctions numpy (rappelons que ces fonctions agissent terme à terme sur les vecteurs), tracer les deux demis-cercles précédents.
On pourra au besoin utiliser l'aide python :

```python
help(np.linspace)
```

**Exercice 2**
On souhaite évaluer l'aire du cercle de rayon 1 en calculant l'intégrale $\displaystyle\int_0^1 \sqrt{1-x^2}\mathrm{d}x$.

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. Créer une fonction `trapeze(g, a, b, n)` qui calcule l'intégrale $\displaystyle\int_a^b g(t)\mathrm{d}t$ par la méthode des trapèzes en utilisant exactement $n$ trapèzes.
   Pour tester votre fonction, on prendra $n=10^4$, la fonction $f$ crée 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

```python
import scipy.integrate as sci
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

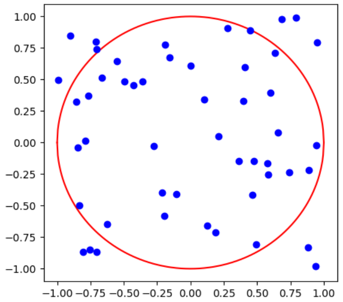
$$\forall x\in\mathbb{R}\ \mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\mathrm{d}t$$

1. Définir en python la fonction `erf` correspondante.
2. Tracer la courbe représentative de `erf` pour $x\in[-3,3]$.
3. Tracer sur le même graphique la courbe de la fonction `scs.erf` obtenue par

```python
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 $x=\pm1,\ y=\pm1$.

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 $\mathcal{C}$ ?
2. Imaginons que l'on choisisse N points $\mathrm{A}_1,\dots,\mathrm{A}_\mathrm{N}$ points dans le carré et que l'on obtienne $n\leqslant\mathrm{N}$ points à l'intérieur du cercle. Quelle quantité est approximée par $\frac{n}{\mathrm{N}}$ ?
3. Pour un point fixé $\mathrm{A}_i=(x_i,y_i)$, comment vérifier si $\mathrm{A}_i$ est à l'intérieur du disque ?
4. 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.

```python
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 $\mathrm{N} = 10^3, 10^4, 10^5, 10^6$. Comparer à la méthode des trapèzes.

# III Pour aller plus loin

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

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

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