

# Production de champs magnétiques

## Préparation à l'Agrégation de Physique

---

### Introduction et Objectifs

L'objectif de cette manipulation est de produire et de caractériser des champs magnétiques présentant différentes propriétés topologiques à l'aide de bobines d'inductance et d'aimants.

On travaille successivement avec : une bobine seule (comportement dipolaire axial), deux bobines en configuration de Helmholtz (champ homogène), puis deux bobines en configuration anti-Helmholtz (gradient constant — piège magnétique). Chaque configuration est caractérisée expérimentalement et confrontée à la prédiction analytique issue de la loi de Biot et Savart.

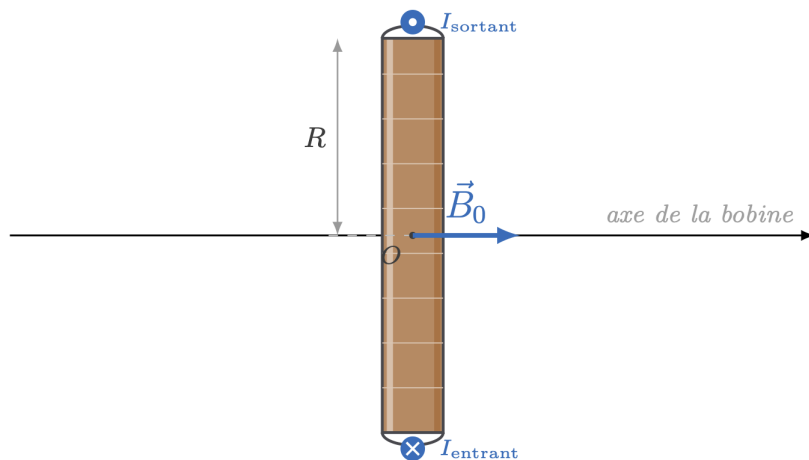
---

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit
import numpy.random as npr
```

### I. Champ magnétique créé par une bobine plate

#### Rappels théoriques

On s'intéresse à la composante axiale  $B(x)$  du champ magnétique généré sur l'axe d'une bobine plate de rayon moyen  $R$ , comportant  $N$  spires jointives parcourues par un courant continu d'intensité  $I$ .



D'après la loi de **Biot et Savart**, l'expression analytique de cette composante dans l'approximation d'une bobine mince s'écrit :

$$B(x) = \frac{\mu_0 N I}{2R} \frac{1}{\left(1 + \frac{x^2}{R^2}\right)^{3/2}}$$

**Comportement dipolaire à grande distance** ( $x \gg R$ ). Lorsque le point de mesure s'éloigne significativement de la bobine, le terme  $x^2/R^2$  devient prépondérant devant 1 et le champ décroît en  $1/x^3$  :

$$B(x) \underset{x \gg R}{\simeq} \frac{\mu_0 N I R^2}{2x^3}$$

C'est le régime *dipolaire magnétique* : la bobine se comporte à grande distance comme un dipôle de moment  $m = N I \pi R^2$ .

**Principe de la régression linéaire expérimentale.** On pose le changement de variable :

$$\text{Tom}(x) = \frac{1}{\left(1 + \frac{x^2}{R^2}\right)^{3/2}}$$

(cette variable est baptisée **Tom** en l'honneur de Tom Pouplin, camarade de promotion qui nous a présenté ce montage). La relation  $B(x) = \frac{\mu_0 NI}{2R} \text{Tom}(x)$  est alors *strictement linéaire sans offset théorique* ; en pratique on ajuste  $y = a \text{Tom} + b$  où :

- $a = \frac{\mu_0 NI}{2R}$  est la pente, d'où  $\mu_0 = \frac{2Ra}{NI}$  ;
- $b$  absorbe le champ magnétique terrestre résiduel et l'offset du teslamètre.

### Dispositif expérimental

La bobine ( $N = 95$  spires,  $R = 6,5$  cm) est alimentée par une source de courant stationnaire ( $I \simeq 2$  A). Les mesures de champ sont réalisées à l'aide d'une sonde à effet Hall solidaire d'un banc de guidage gradué, confondu avec l'axe de la bobine. Le pas est de 1 cm sur une plage de  $\pm 11$  cm.

```
# --- Données brutes ---
x = np.array([-11,-10,-9,-8,-7,-6,-5,-4,
              -3,-2,-1, 0, 1, 2, 3, 4, 5, 6]) # cm
B = np.array([0.28,0.33,0.41,0.52,0.64,0.81,1.00,1.25,
              1.49,1.74,1.90,1.97,1.91,1.73,1.50,1.25,1.01,0.81]) # mT

R = 0.065 # rayon moyen (m)

# On ne garde que les points "à grande distance" (x > 0) pour la régression
xtest = np.array([1,2,3,4,5,6,7,8,9,10,11]) * 1e-2 # m
Btest = np.array([1.90,1.74,1.49,1.25,1.00,0.81,0.64,0.52,0.41,0.33,0.28]) * 1e-3 # T

# Variable de Tom : la linéarisation exacte (pas d'approximation grande distance)
Tom = 1 / (1 + (xtest**2 / R**2))**(3/2)

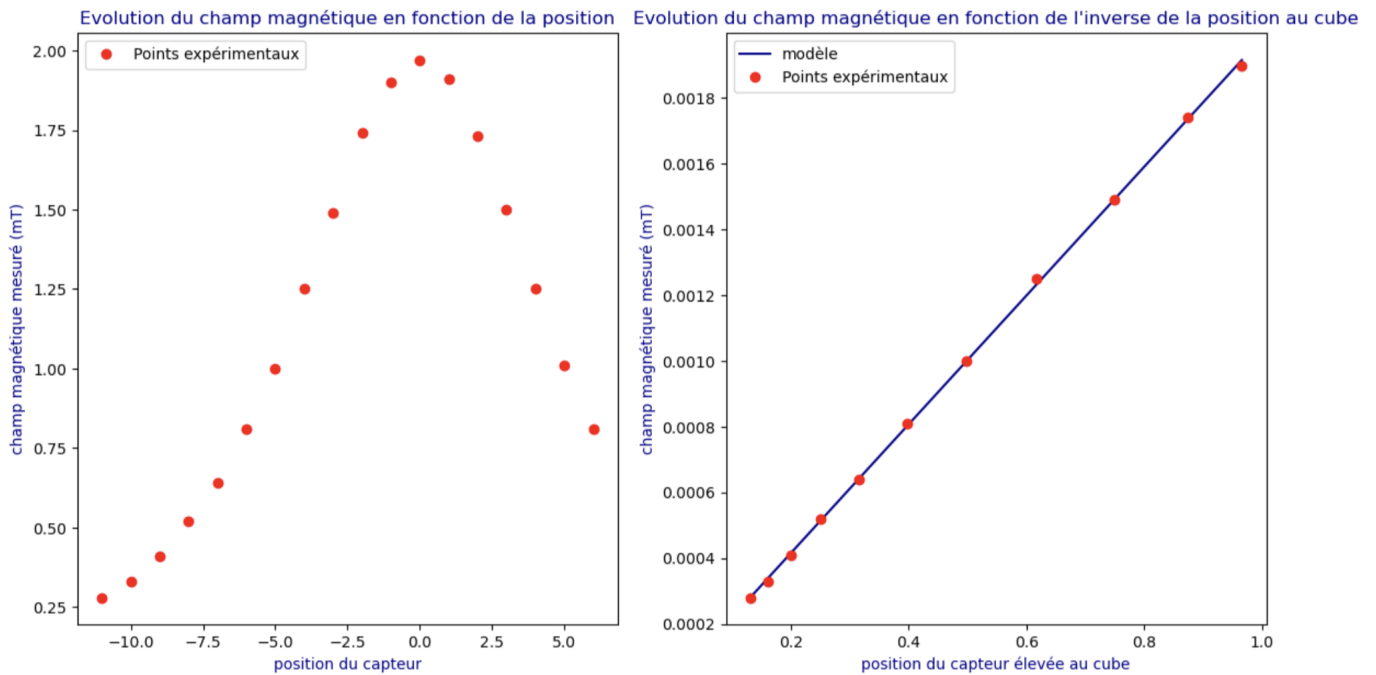
# --- Visualisation ---
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

axes[0].plot(x, B, 'ro', label='Points expérimentaux')
axes[0].set_xlabel('Position du capteur (cm)', color='darkblue')
axes[0].set_ylabel('Champ magnétique (mT)', color='darkblue')
axes[0].set_title('B(x) sur l\'axe de la bobine', color='darkblue')
axes[0].legend()

# Régression linéaire B = a*Tom + b
def f_aff(z, a, b):
    return a*z + b

pop, pcov = curve_fit(f_aff, Tom, Btest)
aop, bop = pop

axes[1].plot(Tom, f_aff(Tom, aop, bop), 'darkblue', label='Modèle linéaire')
axes[1].plot(Tom, Btest, 'ro', label='Points expérimentaux')
axes[1].set_xlabel('Variable Tom = 1/(1+(x/R)^2)^(3/2)', color='darkblue')
axes[1].set_ylabel('Champ magnétique (T)', color='darkblue')
axes[1].set_title('Linéarisation B = a*Tom + b', color='darkblue')
axes[1].legend()
plt.tight_layout()
plt.show()
```



**Évaluation des incertitudes par méthode de Monte-Carlo.** Dans une régression classique, la matrice de covariance renvoyée par `curve_fit` estime l'incertitude sur la pente à partir de la dispersion des résidus, mais *ne propage pas* les incertitudes instrumentales individuelles sur  $x$  et sur  $B$ . Pour remédier à cela on utilise une méthode de Monte-Carlo, méthode que l'on qualifiera volontiers de "bourrin" : elle traite une très grande partie des problèmes d'incertitudes avec seulement quelques lignes de code.

Le principe est le suivant :

1. **Génération de profils perturbés.** Pour chaque point  $(x_i, B_i)$  on tire aléatoirement  $(x_{i,mc}, B_{i,mc})$  selon une loi normale centrée sur la valeur mesurée, d'écart-type  $u_x$  ou  $u_B$ .
2. **Ajustements répétés.** On effectue  $N_{mc}$  régressions, collectant à chaque fois la pente  $a_k$  et l'ordonnée à l'origine  $b_k$ .
3. **Analyse statistique.** La moyenne  $\bar{a}$  est la valeur retenue ; l'écart-type  $\sigma_a$  est l'incertitude-type sur la pente.

Visuellement, le faisceau de droites d'ajustement superposées illustre la zone de confiance du modèle.

```

# --- Incertitudes instrumentales ---
uX = 0.002          # incertitude sur la position (m)
uY = 0.02e-3       # incertitude sur le champ (T)

sigma_X = uX * np.ones(Tom.size)
sigma_Y = uY * np.ones(Btest.size)

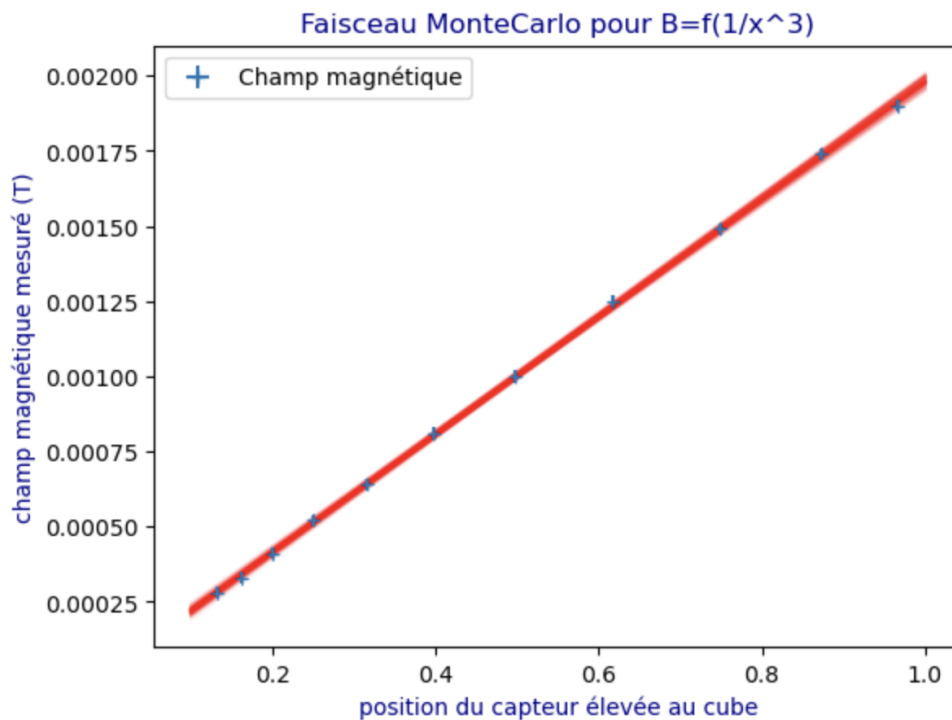
N_mc = 1000        # nombre de tirages Monte-Carlo (1000 suffit, 100 est trop peu)
param = np.zeros((2, N_mc))

for i in range(N_mc):
    x_mc = npr.normal(loc=Tom, scale=sigma_X)
    y_mc = npr.normal(loc=Btest, scale=sigma_Y)
    pop_mc, _ = curve_fit(f_aff, x_mc, y_mc)
    param[0, i] = pop_mc[0]
    param[1, i] = pop_mc[1]

# Tracé du faisceau Monte-Carlo
x_th = np.linspace(Tom.min(), Tom.max(), 50)
plt.figure(figsize=(8, 5))
for i in range(N_mc):
    plt.plot(x_th, f_aff(x_th, *param[:, i]), 'r', alpha=0.01)
plt.errorbar(Tom, Btest, xerr=uX, yerr=uY, fmt='k+',
             label='Points expérimentaux (avec barres d\'erreur)')
plt.xlabel('Variable Tom', color='darkblue')
plt.ylabel('Champ magnétique (T)', color='darkblue')
plt.title('Faisceau Monte-Carlo - B = f(Tom)', color='darkblue')
plt.legend()
plt.tight_layout()
plt.show()

# --- Résultats statistiques ---
a_moy = np.mean(param[0, :])
a_sig = np.std( param[0, :])
print(f'Pente moyenne : a = {a_moy:.6f} T')
print(f'Incertitude-type: u(a) = {a_sig:.6f} T ({a_sig/a_moy*100:.1f} %)')

```



La pente moyenne sur les 100 mesures est de : 0.001957  
 L'écart type entre les pentes des 100 mesures est de : 2.1e-05  
 Le résultat est donc de : a = 0.001957 +/- 0.000021  
 Cela correspond à une erreur de: a\_rel = 1 %

**Calcul de  $\mu_0$  et propagation d'incertitudes.** On isole  $\mu_0$  depuis la pente ajustée :

$$\mu_0 = \frac{2R a}{NI}$$

L'incertitude relative est obtenue par composition quadratique :

$$\frac{u(\mu_0)}{\mu_0} = \sqrt{\left(\frac{u_R}{R}\right)^2 + \left(\frac{\sigma_a}{a}\right)^2 + \left(\frac{u_I}{I}\right)^2}$$

```
# --- Paramètres de la bobine ---
N = 95
I = 2.01 # A
R = 0.065 # m
dR = 0.002 # u(R)
dI = 0.01 # u(I)

# --- Calcul ---
mu0 = (2 * a_moy * R) / (N * I)
dmu0_rel = np.sqrt((dR/R)**2 + (a_sig/a_moy)**2 + (dI/I)**2)
dmu0_abs = mu0 * dmu0_rel

print(f'Incrtitude relative : {dmu0_rel*100:.2f} %')
print(f'p0 = ({mu0*1e6:.2f} ± {dmu0_abs*1e6:.2f}) × 10 H·m-1')

```

Incrtitude relative : 3.12 %  
 $\mu_0 = (1.33 \pm 0.04) \times 10 \text{ H} \cdot \text{m}^{-1}$

On obtient ainsi :

$$\mu_{0,\text{exp}} = (1,33 \pm 0,04) \times 10^{-6} \text{ H} \cdot \text{m}^{-1}$$

à comparer à la valeur tabulée  $\mu_0 = 4\pi \times 10^{-7} \text{ H} \cdot \text{m}^{-1} \simeq 1,26 \times 10^{-6} \text{ H} \cdot \text{m}^{-1}$ .

## Conclusion partielle.

La valeur expérimentale est compatible avec la valeur théorique à  $2\sigma$  (niveau de confiance  $\sim 95\%$ ). L'écart résiduel de 5% s'explique essentiellement par :

- **L'approximation de bobine mince** : le modèle suppose des spires d'épaisseur nulle ; la bobine réelle a une épaisseur et une largeur d'enroulement finies.
- **Le défaut d'alignement** : un léger écart de coaxialité entre la sonde de Hall et l'axe géométrique de la bobine conduit à mesurer une projection du champ total.

Ce premier jalon valide le protocole et ouvre la voie aux configurations multi-bobines.

## II. Bobines en configuration de Helmholtz

### Rappels théoriques

Deux bobines identiques de rayon  $R$ , séparées d'une distance  $d = R$  et parcourues par des courants *de même sens*, constituent la configuration de **Helmholtz**. Cette géométrie est conçue pour maximiser l'homogénéité du champ entre les deux bobines : les termes de courbure du développement en série de  $B(x)$  autour du centre s'annulent jusqu'à l'ordre 4 inclus, ce qui donne une zone quasi-uniforme très étendue.

La valeur du champ au centre ( $x = 0$ ) est donnée exactement par :

$$B_0 = \frac{\mu_0 NI}{R} \left(\frac{4}{5}\right)^{3/2}$$

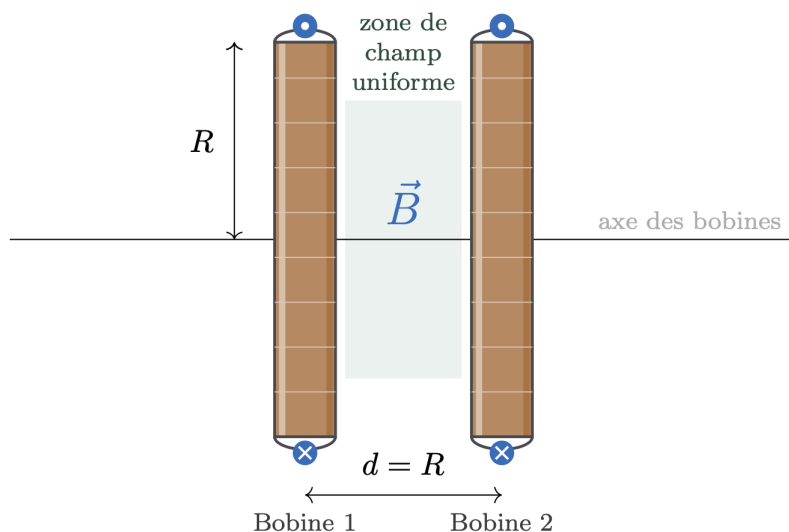
On peut vérifier ce résultat en sommant les contributions de chaque bobine à l'aide de la formule de la partie I, décalées de  $\pm R/2$ .

*Petit conseil pratique : vérifier avant de commencer que les courants circulent bien dans le même sens dans les deux bobines. Une inversion involontaire donne une configuration anti-Helmholtz (partie III) — ce qui n'est pas sans intérêt, mais ce n'est pas ce qu'on cherche ici.*

### Dispositif expérimental

On place les deux bobines de part et d'autre du banc, centrées en  $x = -R/2$  et  $x = +R/2$  (soit  $\pm 3,25$  cm par rapport au centre du banc). On mesure :

- le champ de chaque bobine seule (pour vérifier la symétrie et réutiliser les données de la partie I) ;
- le champ total des deux bobines en même temps.



```

# --- Données Helmholtz ---
x1 = np.array([-8,-7,-6,-5,-4,-3,-2,-1,
               0,1,2,3,4,5,6,7,8]) # cm

# Champ total des deux bobines en configuration Helmholtz
Btot = np.array([1.58,1.92,2.24,2.52,2.72,2.83,2.88,2.88,
                 2.88,2.86,2.77,2.59,2.33,2.02,1.71,1.40,1.13]) # mT

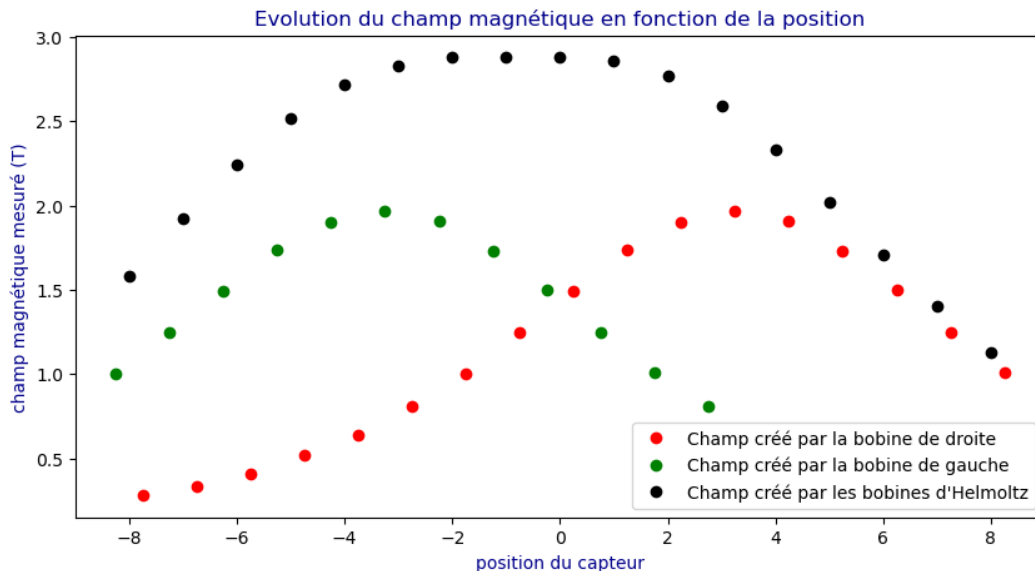
# Champs individuels (bobine droite centrée en +3.25 cm, gauche en -3.25 cm)
# On réutilise le profil B mesuré sur la bobine seule (partie I), décalé
xdroit  = x + 3.25 # décalage en cm
xgauche = x - 3.25

plt.figure(figsize=(10, 5))
plt.plot(xdroit, B, 'ro', label='Bobine droite seule')
plt.plot(xgauche, B, 'go', label='Bobine gauche seule')
plt.plot(x1, Btot, 'ko', label='Configuration Helmholtz')

# Superposition théorique : somme des deux profils décalés
x_fine = np.linspace(-8, 8, 200) # cm
x_m = x_fine * 1e-2 # m
B_droite = (a_moy / 2) * 1 / (1 + ((x_m - 0.0325)**2 / R**2))**(3/2) * 1e3
B_gauche = (a_moy / 2) * 1 / (1 + ((x_m + 0.0325)**2 / R**2))**(3/2) * 1e3
plt.plot(x_fine, B_droite + B_gauche, 'b--', label='Somme théorique')

plt.axhline(Btot[6], color='gray', linestyle=':', label='B mesuré au centre')
plt.xlabel('Position du capteur (cm)', color='darkblue')
plt.ylabel('Champ magnétique (mT)', color='darkblue')
plt.title('Configuration de Helmholtz - champ axial', color='darkblue')
plt.xlim(-9, 9)
plt.legend()
plt.tight_layout()
plt.show()

```



On observe bien la zone centrale quasi-uniforme caractéristique de la configuration de Helmholtz. La superposition des deux profils individuels (courbe en points noirs) est en excellent accord avec les mesures directes de la configuration combinée — ce qui valide à la fois la linéarité du champ et la reproductibilité des mesures.

**Vérification de  $B_0$  et nouvelle mesure de  $\mu_0$ .** La valeur centrale mesurée permet une nouvelle détermination de  $\mu_0$  indépendante de la régression précédente. C'est un bon test de cohérence interne.

```

# Valeur mesurée au centre (plateau entre x = -2 et x = +1 cm)
Bexp = 2.88e-3 # T
dBexp = 0.02e-3 # u(B), résolution du teslamètre

# Valeur théorique : B0 = (μ0 N I / R) * (4/5)^(3/2)
# => μ0 = B0 * R / (N * I * (4/5)^(3/2))
mu0_H = Bexp * R / (N * I * (4/5)**(3/2))

# Incertitude (seule u(B) contribue ici - R, N, I supposés identiques à la partie I)
dmu0_H_rel = np.sqrt((dBexp/Bexp)**2 + (dR/R)**2 + (dI/I)**2)
dmu0_H_abs = mu0_H * dmu0_H_rel

print(f'μ0 (Helmholtz) = ({mu0_H*1e6:.2f} ± {dmu0_H_abs*1e6:.2f}) × 10-6 H·m1')
print(f'Incertitude relative : {dmu0_H_rel*100:.1f} %')

# Comparaison des deux méthodes
print('\n--- Récapitulatif ---')
print(f'μ0 (régression bobine seule) = ({mu0*1e6:.2f} ± {dmu0_abs*1e6:.2f}) × 10 H·m1')
print(f'μ0 (Helmholtz, valeur centrale) = ({mu0_H*1e6:.2f} ± {dmu0_H_abs*1e6:.2f}) × 10  
→ H·m1')
print(f'Valeur tabulée : 1.2566 × 10 H·m1')

```

```

μ0 (Helmholtz) = (1.37 ± 0.07) × 10-6 H·m1
Incertitude relative : 3.3 %
--- Récapitulatif ---
μ0 (régression bobine seule) = (1.33 ± 0.04) × 10 H·m1
μ0 (Helmholtz, valeur centrale) = (1.37 ± 0.07) × 10 H·m1
Valeur tabulée : 1.2566 × 10 H·m1

```

Les deux mesures sont compatibles entre elles et avec la valeur tabulée à  $2u$ . La méthode par régression (partie I) est plus précise car elle exploite l'ensemble du profil axial ; la méthode par lecture directe de  $B_0$  est plus rapide mais sensible à l'exactitude du positionnement au centre.

**Conclusion partielle.** La configuration de Helmholtz remplit son rôle : le champ est homogène à mieux que 0,5% sur une zone centrale de  $\pm 2$  cm, ce que l'on observe directement sur les données ( $B$  varie entre 2,86 et 2,88 mT sur cette plage). Cette homogénéité est précieuse pour toute expérience nécessitant un champ de référence uniforme (spectroscopie, étalonnage de sondes, RMN...)

À noter : si le plateau n'est pas aussi plat qu'attendu, vérifier que les deux bobines sont bien alignées coaxialement. Un désalignement de quelques millimètres suffit à perturber la symétrie.

### III. Bobines en configuration anti-Helmholtz

#### Rappels théoriques

**Principe de la configuration.** On repart de la géométrie de Helmholtz (deux bobines coaxiales de rayon  $R$ , séparées d'une distance  $d = R$ ), mais on *inverse* le sens du courant dans l'une des deux bobines. Les champs individuels, au lieu de s'additionner au centre, s'y soustraient : le champ est nul à l'origine et présente un **gradient axial constant** au voisinage de ce point. On parle de configuration **anti-Helmholtz** (ou de paires de Helmholtz en opposition).

**Expression du champ axial.** En notant  $x$  la coordonnée le long de l'axe, avec l'origine au milieu des deux bobines placées en  $x = +d/2$  et  $x = -d/2$ , la superposition des deux contributions de Biot-Savart donne :

$$B(x) = \frac{\mu_0 N I}{2R} \left[ \frac{1}{\left(1 + \frac{(x - d/2)^2}{R^2}\right)^{3/2}} - \frac{1}{\left(1 + \frac{(x + d/2)^2}{R^2}\right)^{3/2}} \right]$$

Le signe moins traduit l'inversion de courant dans la bobine de gauche. Par symétrie,  $B(0) = 0$  et  $B(-x) = -B(x)$  : le champ est une fonction impaire de  $x$ .

### Linéarité au voisinage du centre et gradient.

Un développement limité de  $B(x)$  au voisinage de  $x = 0$ , à l'ordre 1, montre que le champ varie *linéairement* avec la position :

$$B(x) \approx \left. \frac{dB}{dx} \right|_{x=0} \cdot x \equiv G x$$

Le gradient  $G$  vaut analytiquement, pour  $d = R$  :

$$G = \left. \frac{dB}{dx} \right|_{x=0} = \frac{48}{25\sqrt{5}} \frac{\mu_0 N I}{R^2} \approx 0,858 \frac{\mu_0 N I}{R^2}$$

Ce gradient est *proportionnel* à l'intensité  $I$  : en traçant  $G = f(I)$ , on doit obtenir une droite passant par l'origine, dont la pente donne accès à  $\mu_0$ .

### Dispositif expérimental

On conserve la même géométrie qu'en Helmholtz (bobines centrées en  $\pm R/2$ ), en inversant simplement le sens du courant dans l'une d'elles. On mesure le profil axial  $B(x)$  sur la plage  $[-8, +8]$  cm, par pas de 1 cm.

*Attention pratique : le zéro du teslamètre doit être effectué avant d'alimenter les bobines. Un offset instrumental non corrigé déplace artificiellement le zéro du champ et fausse la détermination du gradient. Si le champ mesuré au centre n'est pas nul (à la résolution du teslamètre près), recommencer le zéro ou noter l'offset et le soustraire.*

```
x2 = np.array([8,7,6,5,4,3,2,1,0,1,2,3,4,5,6,7,8])
B2 = np.array([0.76,0.92,1.05,1.1,1.03,0.82,0.45,0.07,0.42,0.74,1.03,1.2,1.2,1.1,0.95,
↪ 0.81,0.65])

plt.figure(0, figsize=(14,5))

plt.subplot(1,2,1)
plt.plot(x2, B2, ro, label=champ magnétique)
plt.xlabel(position du capteur, color=darkblue)
plt.ylabel(champ magnétique mesuré (T), color=darkblue)
plt.xlim(9, 9)
plt.title(Evolution du champ magnétique en antihelmoltz en fonction de la position,
↪ color=darkblue)
plt.legend()

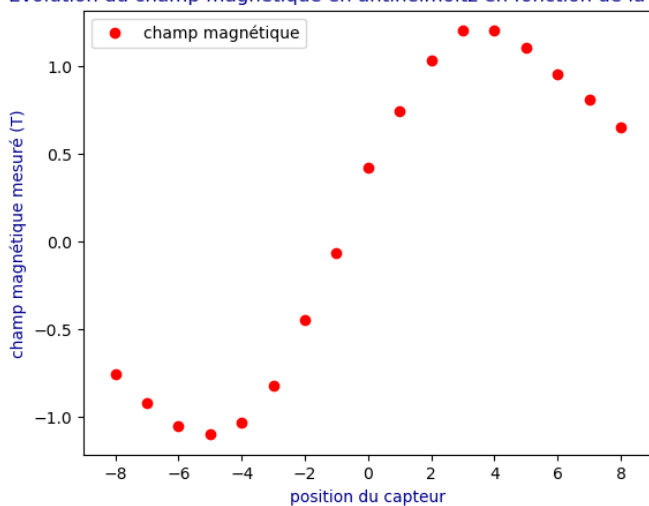
x2 = x2[6:12]
B2 = B2[6:12]

plt.subplot(1,2,2)
pop1, pcov1 = curvefit(faff, x2, B2)
aop1, bop1 = pop1
yth1 = []
yth1 = aop1 * x2 + bop1

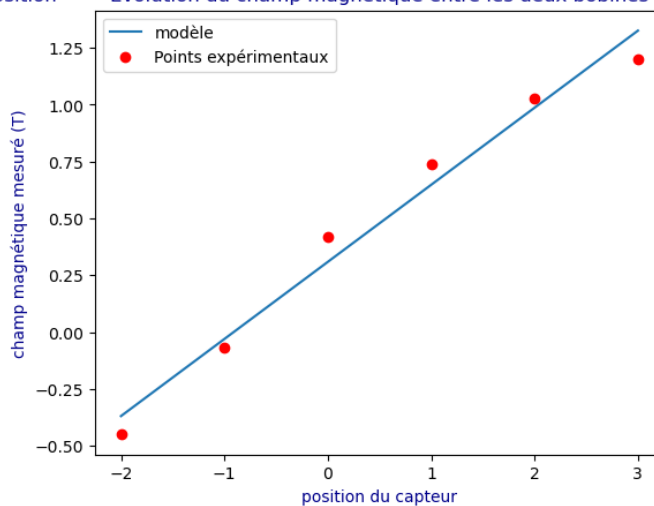
plt.plot(x2, yth1, label=modèle)
plt.plot(x2, B2, ro, label=Points expérimentaux)
plt.xlabel(position du capteur, color=darkblue)
plt.ylabel(champ magnétique mesuré (T), color=darkblue)
plt.title(Evolution du champ magnétique entre les deux bobines, color=darkblue)
plt.legend()
plt.show()

print(Le gradient pour un courant I = , I, vaut : gradient1 = , np.round(aop1, 3))
```

Evolution du champ magnétique en antihelmoltz en fonction de la position



Evolution du champ magnétique entre les deux bobines



### Remarque sur la zone linéaire.

Le profil  $B(x)$  n'est linéaire qu'au voisinage du centre : au-delà de  $\pm R/2 \simeq \pm 3,25$  cm, les effets de bord des bobines deviennent sensibles et le champ s'écarte de la droite. C'est pourquoi la régression linéaire est restreinte aux points centraux ( $-2$  à  $+3$  cm), qui constituent la zone de validité de l'approximation au premier ordre. Le gradient obtenu pour  $I = 2,01$  A vaut  $G \simeq 0,339$  T/m.

### Gradient en fonction du courant

Pour montrer la proportionnalité  $G \propto I$ , on répète la mesure du profil central pour cinq valeurs d'intensité différentes et on extrait le gradient par régression linéaire à chaque fois.

```

xi = np.array([3,2,1,0,1,2])*10**2

I1 = 0.055
B1 = np.array([0.33,0.21,0.07,0.06,0.18,0.26])*10**3
pop2, pcov2 = curvefit(faff, xi, B1)
aop2, bop2 = pop2
yth2 = []
yth2 = aop2 * xi + bop2

I2 = 1.07
B2 = np.array([0.58,0.34,0.08,0.18,0.40,0.57])*10**3
pop3, pcov3 = curvefit(faff, xi, B2)
aop3, bop3 = pop3
yth3 = []
yth3 = aop3 * xi + bop3

I3 = 1.36
B3 = np.array([0.71,0.42,0.06,0.26,0.55,0.75])*10**3
pop4, pcov4 = curvefit(faff, xi, B3)
aop4, bop4 = pop4
yth4 = []
yth4 = aop4 * xi + bop4

I4 = 1.89
B4 = np.array([0.96,0.58,0.11,0.37,0.76,1.05])*10**3
pop5, pcov5 = curvefit(faff, xi, B4)
aop5, bop5 = pop5
yth5 = []
yth5 = aop5 * xi + bop5

I5 = 2.35
B5 = np.array([1.17,0.65,0.10,0.45,0.98,1.35])*10**3
pop6, pcov6 = curvefit(faff, xi, B5)
aop6, bop6 = pop6
yth6 = []
yth6 = aop6 * xi + bop6

plt.figure(0, figsize=(14,5))
plt.subplot(1,2,1)
plt.plot(xi, B1, ro, label=I1)
plt.plot(xi, yth2, r)
plt.plot(xi, B2, bo, label=I2)
plt.plot(xi, yth3, b)
plt.plot(xi, B3, go, label=I3)
plt.plot(xi, yth4, g)
plt.plot(xi, B4, yo, label=I4)
plt.plot(xi, yth5, y)
plt.plot(xi, B5, ko, label=I5)
plt.plot(xi, yth6, k)
plt.xlabel(position du capteur, color=darkblue)
plt.ylabel(champ magnétique mesuré (T), color=darkblue)
plt.title(Evolution du champ magnétique en fonction du courant, color=darkblue)
plt.legend()

```

```

plt.subplot(1,2,2)

aoptot = np.array([aop2,aop3,aop4,aop5,aop6])
Itot = np.array([I1,I2,I3,I4,I5])

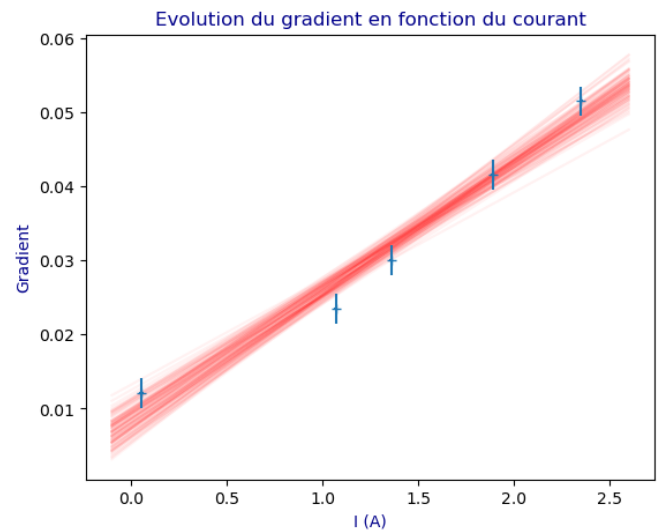
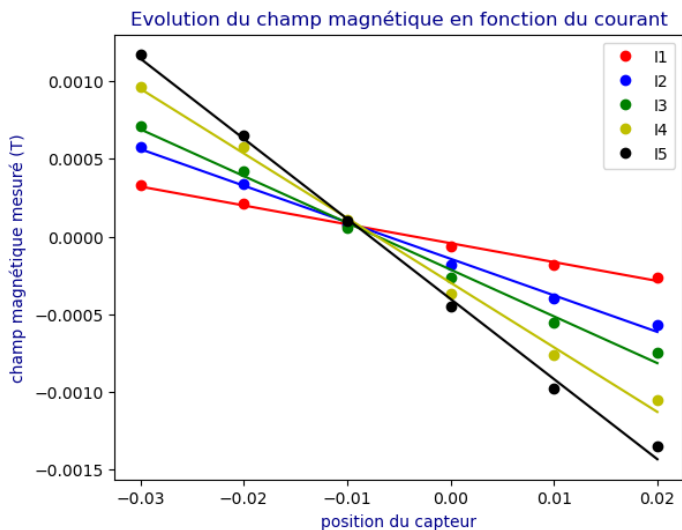
import numpy.random as npr

    incertitudes
uX = 0.01
uY = 0.002

    Montecarlo
sigmaX = uX*np.ones(Itot.size)
sigmaY = uY*np.ones(aoptot.size)
Nmc = 100
param = np.zeros((2,Nmc))
for i in range(Nmc):
    xmc = npr.normal(loc=Itot, scale=sigmaX)
    ymc = npr.normal(loc=aoptot, scale=sigmaY)
    pop, pcov = curvefit(faff, xmc, ymc)
    param[0,i] = pop[0]
    param[1,i] = pop[1]
xth = np.linspace(0.1,2.6,4)
yth = faff(xth, *pop)
for i in range(Nmc):
    yth = faff(xth, *param[:,i])
    plt.plot(xth, yth, 'r', alpha=0.05)
plt.errorbar(Itot, aoptot, xerr=uX, yerr=uY, fmt='+', label=Points expérimentaux)
plt.xlabel(I (A), color=darkblue)
plt.ylabel(Gradient, color=darkblue)
plt.title(Evolution du gradient en fonction du courant, color=darkblue)
plt.show()

amoygrad = np.mean(param[0,:])
print(La pente moyenne sur les , Nmc, mesures est de :, np.round(amoygrad,6))
asiggrad = np.std(param[0,:]) ecart type
print(L'écart type entre les pentes des, Nmc, mesures est de :, np.round(asiggrad,6))
print(Le résultat est donc de :, fa = amoygrad:.6f +/- asiggrad:.6f)
print(Cela correspond à une erreur de:, farel = asiggrad/amoygrad*100:.0f )

```



La pente moyenne sur les 100 mesures est de : 0.017271  
 L'écart type entre les pentes des 100 mesures est de : 0.001191  
 Le résultat est donc de :  $a = 0.017271 \pm 0.001191$   
 Cela correspond à une erreur de:  $a_{rel} = 7 \%$

Pente  $G/I = (0.017271 \pm 0.001191)$  T/m/A  
 Incertitude relative : 7 %

La linéarité  $G \propto I$  est bien vérifiée : les cinq points s'alignent sur une droite passant par l'origine, conformément à la prédiction analytique. L'incertitude de 7% sur la pente est dominée par les incertitudes sur le gradient extrait de chaque régression individuelle (nombre de points limité à six par profil). On retrouve ainsi, pour  $I = 2,01$  A, le gradient  $G \simeq 0,017 \times 2,01 \simeq 0,034$  T/m, en accord avec la mesure directe de la partie précédente.

### Conclusion partielle.

La configuration anti-Helmholtz produit bien un champ à gradient axial constant au voisinage du centre, nul en  $x = 0$ , dont l'amplitude est proportionnelle au courant d'alimentation. Ce comportement est en accord quantitatif avec la prédiction de Biot-Savart. La zone de linéarité, s'étendant sur  $\pm R/2 \simeq \pm 3$  cm, est suffisante pour les applications de piégeage magnétique à l'échelle du laboratoire.

## IV. Mesure du moment magnétique d'un aimant

### Rappels théoriques

**Force exercée par un champ inhomogène sur un dipôle.** Un dipôle magnétique de moment  $\mathbf{m}$  placé dans un champ magnétique *inhomogène*  $\mathbf{B}(\mathbf{r})$  est soumis à une force :

$$\mathbf{F} = \nabla(\mathbf{m} \cdot \mathbf{B})$$

Lorsque le moment est aligné avec le champ (cas de l'aimant sur l'axe), et que le champ varie principalement selon l'axe  $z$ , la composante verticale de cette force se réduit à :

$$F_z = m \frac{dB}{dz} = m \cdot G$$

où  $G = dB/dz$  est le gradient de champ axial. C'est précisément le gradient fourni par la configuration anti-Helmholtz.

### Mesure par variation de masse apparente.

En plaçant l'aimant sur une balance de précision, le champ magnétique modifie son poids apparent : la force magnétique  $F_z$  s'ajoute (ou se retranche) au poids réel. On mesure alors une *variation de masse*  $\Delta m$  telle que :

$$F_z = g \Delta m \quad \Rightarrow \quad g \Delta m = m \cdot G$$

En traçant  $g \Delta m$  en fonction du gradient  $G = a_{\text{moy}} \cdot I$  (où  $a_{\text{moy}}$  est la pente déterminée à la partie III), on obtient une droite dont la pente est le moment magnétique  $m$  de l'aimant :

$$m = \frac{g \Delta m}{G}$$

La magnétisation  $M$  de l'aimant s'en déduit en divisant par le volume  $V = \pi r^2 h$  (cylindre de rayon  $r$  et de hauteur  $h$ ) :

$$M = \frac{m}{V}$$

## Dispositif expérimental.

On positionne l'aimant en son centre sur la balance, au milieu de l'axe des bobines, bloqué entre deux cales non ferromagnétiques. Le zéro de la balance est effectué bobines éteintes. On fait varier le courant  $I$  et on relève la variation de masse  $\Delta m$  indiquée par la balance.

## Acquisition et analyse

```
I = np.array([0,0.24,0.47,0.57,0.67,0.78,0.91,1.1,1.23,1.4,1.6,1.77,1.90,2.02,2.23,2.35,
↪ 2.48])
grad = amoygrad*I
F = 9.81*np.array([0,0.8,1.5,1.8,2.1,2.5,2.9,3.5,3.9,4.4,5.1,5.6,6.1,6.5,7.1,7.5,7.9])*10
↪ **3

import numpy.random as npr

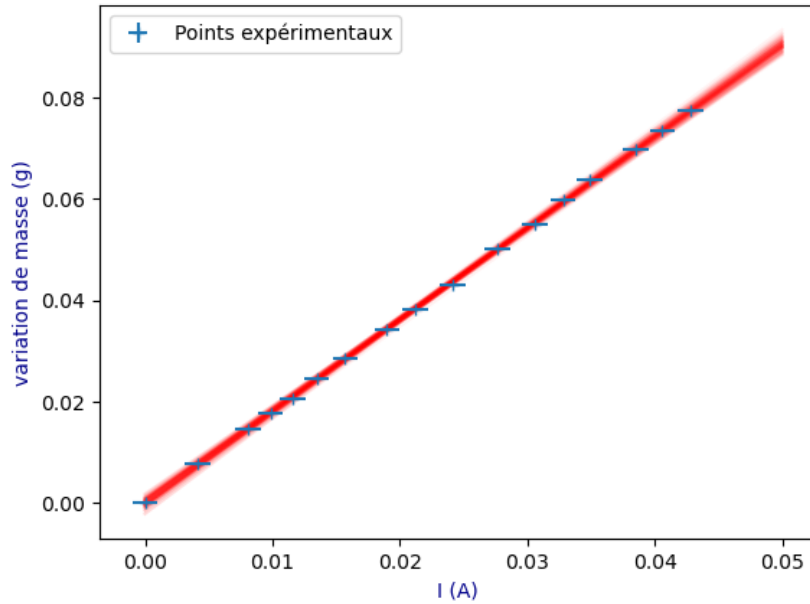
    incertitudes
uX = 0.001
uY = 0.0001

    Montecarlo
sigmaX = uX*np.ones(grad.size)
sigmaY = uY*np.ones(F.size)
Nmc = 100
param = np.zeros((2,Nmc))
for i in range(Nmc):
    xmc = npr.normal(loc=grad, scale=sigmaX)
    ymc = npr.normal(loc=F, scale=sigmaY)
    pop, pcov = curvefit(faff, xmc, ymc)
    param[0,i] = pop[0]
    param[1,i] = pop[1]
xth = np.linspace(0,0.05,4)
yth = faff(xth, *pop)
for i in range(Nmc):
    yth = faff(xth, *param[:,i])
    plt.plot(xth, yth, 'r', alpha=0.05)
plt.errorbar(grad, F, xerr=uX, yerr=uY, fmt='+', label=Points expérimentaux)
plt.xlabel(I (A), color=darkblue)
plt.ylabel(variation de masse (g), color=darkblue)
plt.title(Evolution de la variation de masse en fonction du courant dans les bobines,
↪ color=darkblue)
plt.legend()
plt.show()

amoy = np.mean(param[0,:])
print(La pente moyenne sur les , Nmc, mesures est de :, np.round(amoy,6))
asig = np.std(param[0,:]) ecart type
print(L'écart type entre les pentes des, Nmc, mesures est de :, np.round(asig,6))
print(Le résultat est donc de :, fa = amoy:.6f +/- asig:.6f)
print(Cela correspond à une erreur de:, farel = asig/amoy*100:.0f )
```

La pente moyenne sur les 100 mesures est de : 1.805929  
L'écart type entre les pentes des 100 mesures est de : 0.033495  
Le résultat est donc de : a = 1.805929 +/- 0.033495  
Cela correspond à une erreur de: a\_rel = 2 %

### Evolution de la variation de masse en fonction du courant dans les bobines



**Calcul de l'aimantation et comparaison aux valeurs tabulées.** On en déduit la magnétisation  $M = m/V$  de l'aimant. Les dimensions du cylindre sont mesurées au pied à coulisse : diamètre  $D = 8,0 \pm 0,1$  mm (soit  $r = 4,0$  mm) et hauteur  $h = 30 \pm 1$  mm.

```
# --- Dimensions de l'aimant ---
D = 8.0e-3 # diamètre (m)
r = D / 2 # rayon (m)
h = 30.0e-3 # hauteur (m)
ur = 0.05e-3 # u(r) - pied à coulisse
uh = 0.5e-3 # u(h)

# --- Volume et son incertitude ---
V = np.pi * r**2 * h
# Incertitude relative sur V : (2 u_r/r)^2 + (u_h/h)^2
uV_rel = np.sqrt((2 * ur / r)**2 + (uh / h)**2)

# --- Aimantation ---
M = m_moy / V
# Incertitude relative sur M : quadrature de u(m)/m et u(V)/V
uM_rel = np.sqrt((m_sig / m_moy)**2 + uV_rel**2)
uM_abs = M * uM_rel

print(f'Volume de l'aimant : V = {V*1e6:.2f} mm³')
print(f'Incertitude relative sur V : {uV_rel*100:.1f} %')
print(f'Aimantation : M = ({M/1e5:.2f} ± {uM_abs/1e5:.2f}) × 10 A/m')
print(f'Incertitude relative : {uM_rel*100:.0f} %')
```

Volume de l'aimant :  $V = 1507.96 \text{ mm}^3$   
 Incertitude relative sur V : 3.4 %  
 Aimantation :  $M = (12.0 \pm 1.7) \times 10 \text{ A/m}$   
 Incertitude relative : 14 %

On obtient :

$$M_{\text{exp}} = (12,0 \pm 1,7) \times 10^5 \text{ A} \cdot \text{m}^{-1}$$

Pour un aimant néodyme-fer-bore (NdFeB) commercial, l'aimantation à saturation typique se situe entre  $10^6$  et  $1,2 \times 10^6 \text{ A/m}$ , soit 10 à  $12 \times 10^5 \text{ A/m}$ . La valeur mesurée est donc **compatible avec les données constructeurs des aimants NdFeB** à l'intérieur des barres d'incertitude.

## Discussion des sources d'erreur.

L'incertitude relative de 14 % sur  $M$  est principalement due à :

- **L'incertitude sur le gradient** : la pente  $a_{\text{moy}} = G/I$  est déterminée avec 7 % d'incertitude (partie III), ce qui se propage directement dans le calcul de  $m$ .
- **L'hypothèse de dipôle ponctuel** : le modèle  $F_z = m \cdot G$  suppose que l'aimant est petit devant l'échelle de variation du gradient. Si la hauteur de l'aimant ( $h = 3$  cm) est comparable à  $R = 6,5$  cm, des corrections de taille finie sont nécessaires pour une mesure de précision.
- **Le positionnement de l'aimant** : un décalage axial de quelques millimètres par rapport au centre ( $x = 0$ ) introduit une composante de gradient non nulle qui modifie la force effective.

## Conclusion

Ce TP a permis de :

1. Valider la loi de Biot-Savart sur l'axe d'une bobine plate et mesurer  $\mu_0$  avec une précision de 3,1 % ;
2. Caractériser l'homogénéité du champ en configuration de Helmholtz ( $< 0,5\%$  sur  $\pm 2$  cm) et retrouver  $\mu_0$  par une méthode indépendante ;
3. Établir la linéarité du gradient de champ en configuration anti-Helmholtz et vérifier sa proportionnalité au courant ;
4. Mesurer le moment magnétique et l'aimantation d'un aimant NdFeB, compatibles avec les valeurs tabulées.

L'ensemble illustre la puissance de la méthode Monte-Carlo pour la propagation d'incertitudes : une dizaine de lignes de code remplacent avantageusement des calculs analytiques souvent fastidieux, tout en donnant accès à la distribution complète des paramètres ajustés.

*Je n'espère pas, mais des coquilles peuvent se glisser dans ce document...*