F6. Montrer à l'aide du résultat de la question **C6** que la force électromotrice $e_{spire}(y,t)$ qui apparaît aux bornes de la spire d'abscisse y de la bobine de mesure a pour expression :

$$e_{spire}(y,t) = 6bH_0\omega\sin(2\omega t) \iint_{S(y,x>0)} H_m^2(x,y,z)\cos^2(\theta(x,y))dxdz \tag{6}$$

où $S(x>0,y)$ est la section du tore définie précédemment.

F7. Déterminer l'expression du nombre de spires dN de la bobine situées entre y et $y+dy$. À l'aide de la question **F6**, en déduire l'expression de la force électromotrice de induite aux bornes de ces dN spires.

F8. Montrer que la force électromotrice $e(t)$ qui apparaît aux bornes de la bobine de mesure s'écrit :

$$e(t) = 3\,\alpha\, b\,\frac{N_2N_1^2I_m^2}{8\pi^2}\frac{H_0}{R+a}\omega\,\sin(2\omega t) \tag{7}$$

où $\alpha = \iiint_V \frac{\cos^2\theta}{r^2}dV$ et V est le volume du noyau torique.

F9. En l'absence d'excitation magnétique extérieure, que vaut la tension induite aux bornes du bobinage de mesure ? Quel avantage présente le magnétomètre à noyau torique par rapport au magnétomètre à barreau rectiligne ?

F10. Indiquer comment mesurer simultanément l'intensité du champ magnétique dans deux directions orthogonales avec un seul magnétomètre à noyau torique.

Étude numérique

On se propose d'étudier quelques aspects associés au traitement numérique du signal d'un magnétomètre à vanne de flux.
On utilisera uniquement les fonctions données en annexe. Les bibliothèques correspondantes seront supposées avoir été préalablement importées sous Python. L'usage d'autres fonctions sur les listes telles que `max(liste)`, `min(liste)` ou encore `sum(liste)` est interdit. Ces fonctions devront être programmées explicitement si nécessaire.

G / Quelques aspects algorithmiques associés au traitement des données

*Dans l'étude du magnétomètre torique effectuée dans la partie **F**, on est amené à évaluer l'intégrale* $\alpha = \iiint_V \frac{\cos^2\theta}{r^2}dV$ *(question **F8**). On cherche à déterminer numériquement la valeur de* α.

G1. Écrire une fonction `f(x,y,z)` recevant pour paramètres les coordonnées x, y et z d'un point M et qui renvoie la valeur $\frac{\cos^2\theta}{r^2}$.
On pourra noter que $\frac{\cos^2\theta}{r^2} = \frac{x^2}{(x^2+y^2)^2}$.

G2. Écrire une fonction `Choisir(a,b)` qui renvoie une valeur choisie aléatoirement de manière uniforme entre les bornes a et b passées en argument.

*On propose de déterminer la valeur de $\alpha = \iiint_V f(M)dV$ par une méthode numérique appelée intégration Monte Carlo. L'intégrale s'effectue sur le tore de volume V correspondant au noyau de la bobine ferromagnétique étudiée dans la partie **F**.*
L'intégration Monte Carlo est basée sur le fait que l'intégrale de f(M) sur le volume V peut être approximée par $\alpha \simeq V\langle f\rangle$ avec $\langle f\rangle = \frac{1}{N}\sum_{i=1}^{N} f(M_i)$ où les N points M_i sont choisis aléatoirement et uniformément à l'intérieur du volume V.
L'écart typique entre la valeur exacte de l'intégrale $\iiint_V f(M)dV$ et $V\langle f\rangle$ est donné par $V\sqrt{\frac{\langle f^2\rangle - \langle f\rangle^2}{N}}$ avec $\langle f^2\rangle = \frac{1}{N}\sum_{i=1}^{N} f(M_i)^2$.
Quand le nombre N est suffisamment grand, $V\langle f\rangle$ tend vers $\iiint_V f(M)dV$.
On donne un algorithme permettant de déterminer l'intégrale α effectuée sur le volume V, ainsi que l'erreur relative sur α. On suppose que le nombre de points N est une variable globale préalablement fixée.

- *Initialiser les variables I et I2 à 0.*
- *Effectuer les opérations suivantes jusqu'à obtenir N points dans le tore :*
  - *Choisir un point M au hasard dans le parallélépipède rectangle (de volume minimal) contenant le tore.*
  - *Si le point M est dans le tore, on détermine $f(M)$. On ajoute cette valeur à I, et la valeur de $f(M)^2$ à la variable I2.*
- *Renvoyer l'estimation de la valeur de α et l'erreur relative associée.*

On considérera le parallélépipède rectangle défini par :

$$-(R+a) \leq x \leq R+a, \qquad -(R+a) \leq y \leq R+a \qquad \text{et} \qquad -a \leq z \leq a.$$

On pourra noter qu'un point M de coordonnées cartésiennes (x,y,z) appartient au tore si $\left(\sqrt{x^2+y^2} - R\right)^2 + z^2 \leq a^2$.

G3. Écrire une fonction `Est_dans_tore(x,y,z)` recevant pour paramètres les coordonnées x, y et z et qui renvoie `True` si le point M est dans le tore et `False` sinon. Les rayons R et a caractérisant le tore seront supposés être des variables globales.

G4. Écrire une fonction `Choix_point()` qui renvoie les coordonnées x, y et z d'un point choisi au hasard dans le parallélépipède rectangle de volume minimal contenant le tore.

G5. Implémenter le calcul de l'intégrale α et de l'erreur relative associée par la méthode Monte Carlo décrite ci-dessus.

*Les différentes opérations décrites dans les parties **D** à **E** du problème peuvent être implémentées de façon numérique. On considère que le signal $u_s(t)$ a été numérisé avec une période d'échantillonnage T_e. Les valeurs numériques associées au signal sont disponibles dans une liste u de taille N. La valeur $u_s(kT_e)$ du signal à l'instant kT_e correspond au kième élément de la liste u.*

L'opération de multiplication apparaissant dans la chaîne de traitement de données (figure 8) peut être implémentée numériquement.

G6. Écrire une fonction `Produit(sx,sy)` qui renvoie le produit des deux listes `sx` et `sy` passées en argument. Les deux listes seront supposées être de même longueur.

*On note s la liste obtenue. On a vu dans la partie **E** que l'information utile correspond à la moyenne de s.*

G7. Écrire une fonction `Moyenne(L)` qui renvoie la valeur moyenne des éléments de la liste `L` passée en argument.

La moyenne m de la liste s est liée à l'intensité du champ magnétique à mesurer. Elle dépend aussi du déphasage entre les signaux sx et sy. La valeur de m est maximale quand le déphasage est nul. Il est donc intéressant de chercher à annuler le déphasage entre les signaux sx et sy avant de procéder à l'étape de multiplication.
Pour cela, on commence par rechercher le décalage temporel entre les deux signaux sx et sy de même période. Ce décalage peut être déterminé à l'aide de la fonction d'inter-corrélation C(n) définie numériquement par :

$$C(n) \;=\; \sum_{k=n}^{M+n} sx_k \cdot sy_{k-n} \tag{8}$$

où M définit la taille de la fenêtre pour effectuer le calcul. L'idée est de rechercher le retard (ou translation nTe) qui permet de maximiser la ressemblance entre les signaux sx et sy. On cherche alors la plus petite valeur de n pour laquelle $C(n)$ est maximale. La fonction $C(n)$ est elle-même définie pour n compris entre 0 et N-1-M où N est le nombre d'éléments dans la liste sx (ou sy).

On souhaite déterminer la fonction d'inter-corrélation des signaux sx et sy supposés de même longueur $N = 2\ell$ et de moyenne nulle.

G8. Proposer une fonction `Valeur_max(L)` qui renvoie la valeur maximale de la liste `L` passée en argument.

G9. On souhaite que les listes `sx` et `sy` soient de moyenne nulle, ce qui n'est pas a priori le cas. Écrire une fonction `Moyenne_nulle(L)` qui modifie la liste `L` passée en argument de façon à ce que sa moyenne soit nulle.

G10. On choisit de calculer la fonction d'inter-corrélation pour $M = \ell$. Écrire la fonction `Intercorr(sx,sy)` qui renvoie la liste contenant les valeurs de la fonction d'inter-corrélation C(n) pour n compris entre 0 et $\ell - 1$.

G11. Quelle est la longueur de la liste renvoyée par la fonction `Intercorr(sx,sy)` ? Quelle est la complexité temporelle associée à l'exécution de cette fonction dans les conditions précisées ici ?

DONNÉES

- Volume d'un tore de rayon R et de section circulaire de rayon a : $V = 2\pi^2 R a^2$

Constantes physiques :

- Perméabilité du vide : $\mu_0 = 4\pi \cdot 10^{-7}\,\mathrm{H \cdot m^{-1}}$
- Permittivité du vide : $\epsilon_0 = 8,854 \cdot 10^{-12}\,\mathrm{F \cdot m^{-1}}$

Formulaire mathématique :

- $\sin^2(x) = \frac{1}{2} - \frac{1}{2}\cos(2x)$
- $\sin^3(x) = -\frac{1}{4}\sin(3x) + \frac{3}{4}\sin(x)$
- $\cos\, a \cdot \cos\, b = \frac{1}{2}\left(\cos(a+b) + \cos(a-b)\right)$

ANNEXE

Fonctions et constantes autorisées - Python	
`len(L)`	renvoie le nombre d'éléments de la liste L.
`L = []`	création d'une liste vide.
`L = [x]*n`	création d'une liste de n éléments ayant tous la valeur contenue dans la variable x.
`L = [x for i in range(n)]`	même opération que précédemment (alternative avec une compréhension de liste).
`L.append(elt)`	ajoute l'élément elt à la liste L. La méthode `append` ne renvoie aucune valeur.
`cos(x)`	renvoie la valeur du cosinus de x radians.
`sin(x)`	renvoie la valeur du sinus de x radians.
`tan(x)`	renvoie la valeur de la tangente de x radians.
`sqrt(x)`	renvoie la racine carrée de x.
`pi`	constante dont la valeur est $\pi = 3.141592...$ à la précision disponible.
`random()`	renvoie un nombre pseudo-aléatoire compris entre 0 et 1 avec une densité de probabilité uniforme.

Fin de l'épreuve

NE RIEN ÉCRIRE DANS CE CADRE

G / Quelques aspects algorithmiques associés au traitement des données

G1. Écrire une fonction `f(x,y,z)` recevant pour paramètres les coordonnées x, y et z d'un point M et qui renvoie la valeur $\frac{\cos^2\theta}{r^2}$.
On pourra noter que $\frac{cos^2\theta}{r^2} = \frac{x^2}{(x^2+y^2)^2}$.

```
def f(x,y,z):
     return x*x/((x*x+y*y)**2)
```

G2. Écrire une fonction `Choisir(a,b)` qui renvoie une valeur choisie aléatoirement de manière uniforme entre les bornes a et b passées en argument.

```
def Choisir(a,b):
      return (b-a)*random()
```

G3. Écrire une fonction `Est_dans_tore(x,y,z)` recevant pour paramètres les coordonnées x, y et z et qui renvoie `True` si le point M est dans le tore et `False` sinon. Les rayons R et a caractérisant le tore seront supposés être des variables globales.

```
def Est_das_tore(x,y,z):
        global R,a
        return ((x*x+y*y)**0.5-R)**2+z**2<=a*a
```

NE RIEN ÉCRIRE DANS CE CADRE

G4. Écrire une fonction `Choix_point()` qui renvoie les coordonnées x, y et z d'un point choisi au hasard dans le parallélépipède rectangle de volume minimal contenant le tore.

```
def Choix_point():
      x=Choisir(-R-a,R+a)
      y=Choisir(-R-a,R=a)
      z=Choisir(-a,a)
      return x,y,z
```

G5. Implémenter le calcul de l'intégrale α et de l'erreur relative associée par la méthode Monte Carlo décrite ci-dessus.

```
I=0
I2=0
for i in range(N):
      x,y,z=Choix_point()
      while Est_dans_tore(x,y,z)==False:
            x,y,z=Choix_point()
      I=I+f(x,y,z)
      I2=I2+f(x,y,z)**2
V=2*pi*pi*R*a*a
alpha=V*I/N
erreur=V*((I2/N-(I/N)**2)/N)**0.5
```

NE RIEN ÉCRIRE DANS CE CADRE

G6. Écrire une fonction `Produit(sx,sy)` qui renvoie le produit des deux listes `sx` et `sy` passées en argument. Les deux listes seront supposées être de même longueur.

```
def produit(sx,sy):
    res=[]
    for i in range(len(sx))
         res.append(sx[i]*sy[i])
    return res
```

G7. Écrire une fonction `Moyenne(L)` qui renvoie la valeur moyenne des éléments de la liste `L` passée en argument.

```
def Moyenne(L):
    s=0
    for a in L:
        s=s+a
    return s/len(L)
```

NE RIEN ÉCRIRE DANS CE CADRE

G8. Proposer une fonction `Valeur_max(L)` qui renvoie la valeur maximale de la liste `L` passée en argument.

```
def Valeur_max(L)
    res=L[0]
    for a in L:
        if a>res:
            res=a
    return res
```

G9. On souhaite que les listes `sx` et `sy` soient de moyenne nulle, ce qui n'est pas a priori le cas. Écrire une fonction `Moyenne_nulle(L)` qui modifie la liste `L` passée en argument de façon à ce que sa moyenne soit nulle.

```
def Moyenne_nulle(L)
    m=Moyenne(L)
    res=[]
    for a in L:
        res.append(a-m)
    return res
```

NE RIEN ÉCRIRE DANS CE CADRE

G10. On choisit de calculer la fonction d'inter-corrélation pour $M = \ell$. Écrire la fonction `Intercorr(sx,sy)` qui renvoie la liste contenant les valeurs de la fonction d'inter-corrélation C(n) pour n compris entre 0 et $\ell - 1$.

```
def Intercorr(sx,sy)
    res=[]
    for n in range(l):
        c=0
        for k in range(l,2*l ):
              c=c+sx[k]*sy[k-n]
        res.append(c)
    return res
```

G11. Quelle est la longueur de la liste renvoyée par la fonction `Intercorr(sx,sy)` ? Quelle est la complexité temporelle associée à l'exécution de cette fonction dans les conditions précisées ici ?

Il y a l éléments dans le résultats.

De plus la première boucle for fait l tours de boucles
et la deuxième aussi pour une multiplication
Il y a donc l² multiplications dans la fonction.
Cela fait une complexité en O(l²)