

Python for dummies : problème 4

Un petit aimant au dessus d'une bobine !

Une étrange histoire magnétique !

Nous allons créer un modèle numérique simple pour prédire la tension induite dans une bobine lorsqu'on déplace un aimant cylindrique au dessus d'une bobine.

L'aimant est un cylindre de rayon $R = 1.25$, d'épaisseur $e = 0.6$ et est situé à une hauteur $z = 0.5$ au dessus d'une bobine de rayon $R = 1$. Toutes les données sont en centimètres. On définit l'origine du système d'axe au centre de la bobine. La bobine et l'aimant sont placés horizontalement avec un vecteur normal $[0, 0, 1]$. Il y a 200 spires dans la bobine.

Le moment magnétique de l'aimant est défini par :

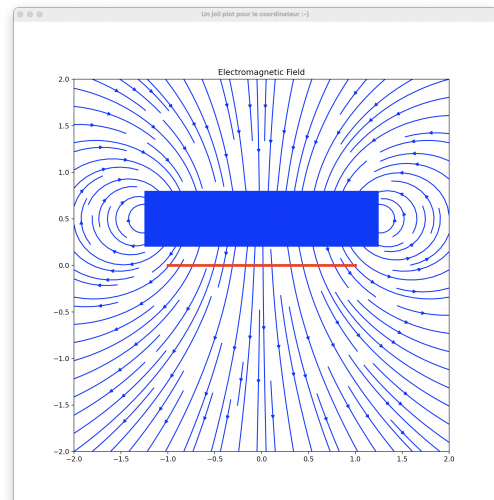
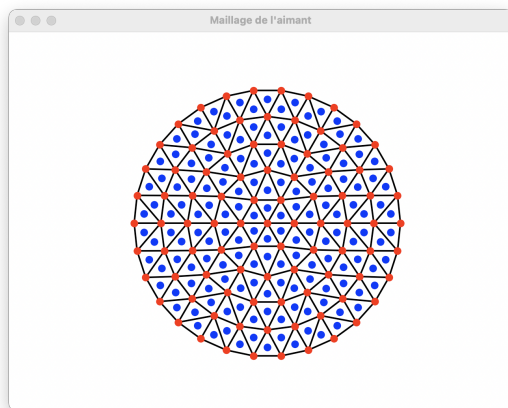
$$\boldsymbol{\mu} = [0, 0, -\mu]$$

La valeur de μ est déduite de la magnétisation résiduelle de Néodyme de bore, de la perméabilité du vide et du volume de l'aimant, comme mon ami Claude vous l'a expliqué.

Et comme votre enseignant favori de physique vous l'a également appris, le champ magnétique $\mathbf{B}$ généré par un dipôle à l'origine à un endroit $\mathbf{r}$ de l'espace est défini par la relation suivante.

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi r^3} \left[3(\boldsymbol{\mu} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} - \boldsymbol{\mu} \right]$$

Pour obtenir le champ magnétique généré par l'aimant, nous allons décomposer celui en certains nombres de triangles ne se recouvrant pas. Il s'agit de créer un maillage de n triangles qui formera une partition du disque de l'aimant en une série de petits triangles Ω_e : ce sont des éléments finis :-). Les **points rouges** sont les **sommets** et les **points bleus** sont les centres de gravité de chaque triangle.



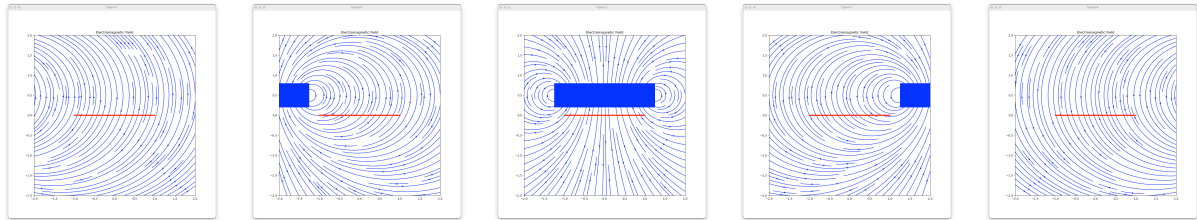
Le champ magnétique de l'aimant sera obtenu en superposant tous les champs magnétiques de dipôles élémentaires μ_e que l'on va placer au centre de gravité de chaque triangle Ω_e .

Comment obtenir le moment magnétique de ces dipôles ? Nous allons procéder de manière très simple en distribuant le moment magnétique de l'aimant proportionnellement à la surface de chaque triangle.

On obtiendra donc évidemment
$$\mu = \sum_{e=1}^n \mu_e.$$

C'est de cette manière que nous avons représenté le champs magnétique avec `python`. Enfin, notre implémentation est presque correcte, on s'est contenté d'écrire $\mu_e = \mu$, mais cela n'a quasiment aucun impact sur les lignes de champ magnétique.

Le processus de translation de l'aimant est illustré ci-dessous dans le plan xz : l'aimant se déplace d'une position $X_{start} = -0.5$ vers une position $X_{stop} = 0.5$. On y observe les lignes du champ magnétique $\mathbf{B}$ créé par la présence de l'aimant et il s'agira donc de calculer le flux de ce champ au travers de la bobine rouge fixe.



L'objectif de ce devoir est maintenant de calculer le flux magnétique qui passe par la bobine lors du déplacement horizontal de l'aimant. Pour réaliser ce calcul, nous allons également décomposer le disque que forme une spire de la bobine avec un maillage de triangles. Et pour faire simple, nous allons utiliser exactement le même maillage que pour l'aimant en ne modifiant que les coordonnées des noeuds.

A chaque instant t du déplacement de l'aimant, on va intégrer numériquement le flux magnétique intercepté par une spire Ω_b de la manière suivante.

$$\phi(t) = \int_{\Omega_b} \mathbf{B}(\mathbf{x}) \cdot \mathbf{e}_z \, d\mathbf{x} = \int_{\Omega_b} B_z(\mathbf{x}) \, d\mathbf{x} = \sum_{i=0}^n \int_{\Omega_i} B_z(\mathbf{x}) \, d\mathbf{x}$$

En utilisant une méthode de Hammer avec un point :-)

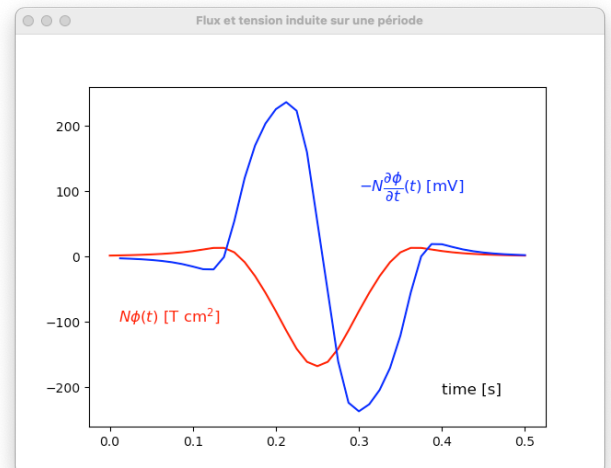
$$\approx \sum_{i=0}^n B_z(\mathbf{X}_i) \Delta_i$$

où $\mathbf{X}_i$ est le centre de gravité et Δ_i la surface de chaque triangle Ω_i du maillage qui forme une partition de Ω_b . On obtient aisément les coordonnées du centre de gravité du triangle en faisant la moyenne des trois sommets.

Ensuite, on déduit la tension induite dans la bobine grâce à la loi de Lenz-Faraday :

$$V_L = -N \frac{d\phi}{dt}(t)$$

où N est le nombre de spires, toujours comme l'a expliqué avec brio notre super enseignant de physique. Le résultat que vous devriez obtenir devrait ressembler à ce qui est illustré. Du moins, si Nicolas et moi ne nous sommes pas trompés :-)



Plus précisément, on vous demande de :

1. Ecrire une fonction

```
phi = magnetComputeInduction(Xmagnet,Ymagnet,Zmagnet,Xcoil,Ycoil,triangles,
                              Xshift,mu0,mu)
```

qui calcule le flux intercepté par une spire pour une série de positions de l'aimant défini par le vecteur **Xshift** de taille **m**. Les autres arguments sont les coordonnées en x , et y des sommets des maillages de l'aimant et de la bobine : **Xmagnet**, **Ymagnet**, **Xcoil**, **Ycoil**. Il s'agit de tableaux qui ont une taille **nNode** égale au nombre de sommets des deux maillages. On fournit également le tableau d'appartenance commun aux deux maillages contenant les indices des 3 sommets de chaque triangle du maillage : **triangles**. C'est un tableau de taille **nElem**×3. Les derniers arguments sont trois réels : les deux paramètres matériels **mu0** et la hauteur de l'aimant par rapport à la bobine **Zmagnet**.

Il faut ensuite renvoyer un tableau de même taille que **Xshift**.

2. Comme d'habitude, pour tester votre programme, on vous a fourni un tout petit programme simple **magnetTest.py** pour tester votre programme. On y construit d'abord le maillage de triangles nécessaire pour notre intégration numérique et pour la superposition des champs magnétiques de tous les dipôles composant notre aimant. Et ensuite, on fait appel à votre fonction !

Le programme fait aussi plein de jolis dessins et d'animation qui seront très utiles pour faire un super joli rapport pour votre projet :-)

```
from numpy import *
from scipy.spatial import Delaunay

mu0      = 4e-7*pi*1e-2      # permeabilité du vide en [H/cm]
Rmagnet  = 1.25              # rayon de l'aimant [cm]
Hmagnet  = 0.6               # épaisseur de l'aimant [cm]
Zmagnet  = 0.5               # position verticale de l'aimant en [cm]
Br       = 1.4               # magnetisation residuelle du Néodyme fer bore (NdFeB) en [T] ou [kg/(A s)]
mu       = Rmagnet**2*Hmagnet*pi*Br / mu0
                                # moment magnétique de l'aimant [A cm2]
Rcoil    = 1                 # rayon de la bobine [cm]
nSpires  = 200

nR       = 6
nTheta  = 6
nNode    = 1 + sum(arange(1,nR))*nTheta
```

```

R      = zeros(nNode)
Theta = zeros(nNode)
index = 1; dR = 1.0/(nR-1)
for i in range(1,nR):
    dTheta = 2*pi/(i*nTheta)
    for j in range(0,i*nTheta):
        R[index] = i*dR
        Theta[index] = j*dTheta; index += 1
X      = R*cos(Theta)
Y      = R*sin(Theta)
triangles = Delaunay(stack((X,Y),1)).simplices
nElem = len(triangles)

print(" Number of triangles : %d " % nElem)
print(" Number of nodes      : %d " % nNode)

m      = 41
Xstart = -5          # [cm]
Xstop  = 5           # [cm]
Xshift = linspace(Xstart,Xstop,m)
Tstart = 0           # [s]
Tstop  = 0.5         # [s]
T,delta = linspace(Tstart,Tstop,m,retstep=True)

Xmagnet = Rmagnet*R*cos(Theta)
Ymagnet = Rmagnet*R*sin(Theta)
Xcoil   = Rcoil*R*cos(Theta)
Ycoil   = Rcoil*R*sin(Theta)

phi      = magnetComputeInduction(Xmagnet,Ymagnet,Zmagnet,Xcoil,Ycoil,triangles,
                                   Xshift,mu0,mu)

phi      = phi * nSpires
voltage = - diff(phi) / (delta*10)

```

3. Ce devoir n'est pas particulièrement compliqué.
Mais il faut essayer d'implémenter ce calcul de manière efficace et surtout d'avoir la bonne réponse !
4. Votre fonction (avec les éventuelles sous-fonctions que vous auriez créées) sera soumise sur **zouLab** sans y adjoindre le programme de test fourni ! Cette fonction devra être soumise via le web : ce travail est individuel et sera évalué. **Pour rappel, toutes vos soumissions seront systématiquement analysées par un logiciel anti-plagiat. Faites vraiment votre programme seul... :-)**
5. A la fin du devoir, il vous suffira de changer les paramètres de l'intervalle temporel et du mouvement entre la bobine et l'aimant pour faire correspondre votre prédiction à votre prototype du projet. Et alors, vous verrez le sourire de notre coordinateur de quadrimestre¹ s'illuminer de bonheur ! Et oui, la modélisation physique et les méthodes numériques sont les clés qui permettent de comprendre ce qu'on observe dans le laboratoire !

¹ Oui, ce fameux enseignant de physique dont personne ne prononce correctement le nom...