

Lentille magnétique

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

```
[2]: m=9.1e-31
e=1.6e-19
B0=40e-3
a=1e-3
def Br(r,z):
    return 3*z*r*B0/(2*a**2*(1+z**2/a**2)**2.5)
def Bz(r,z):
    return B0/(1+z**2/a**2)**1.5
```

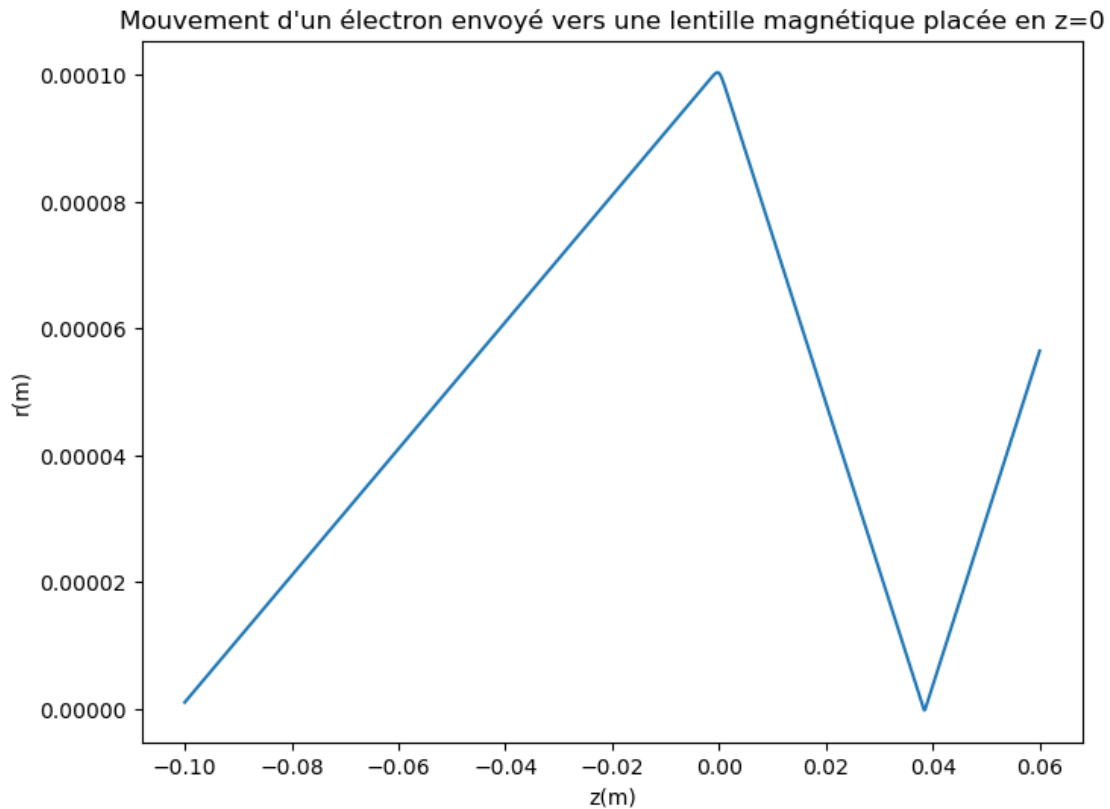
$$\frac{d}{dt} \begin{pmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \\ z \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \dot{r} \\ -\frac{e}{m} r \dot{\theta} B_z(r, z) + r \dot{\theta}^2 \\ \dot{\theta} \\ \frac{e}{mr} \dot{r} B_z(r, z) - \frac{e}{mr} \dot{z} B_r(r, z) - \frac{2\dot{r}\dot{\theta}}{r} \\ \dot{z} \\ \frac{e}{m} r \dot{\theta} B_r(r, z) \end{pmatrix}$$

```
[10]: r0=a/1000
z0=-100*a
v0=2e7
alpha=0.001
def derivee(inconnues,t): # inconnues=[r,theta,z,rpoint,thetapoint,zpoint]
    r,theta,z=inconnues[0],inconnues[1],inconnues[2]
    rpoint,thetapoint,zpoint=inconnues[3],inconnues[4],inconnues[5]
    if r<a*1e-4: r=a*1e-4 ## cette ligne sert à éviter les problèmes en r=0
    rpointpoint=-e/m*r*thetapoint*Bz(r,z)+r*thetapoint**2
    thetapointpoint=(e/m*rpoint*Bz(r,z)-e/m*zpoint*Br(r,z)-2*rpoint*thetapoint)/
    ↪ r
    zpointpoint=e/m*r*thetapoint*Br(r,z)
    return [rpoint,thetapoint,zpoint,rpointpoint,thetapointpoint,zpointpoint]
ci=[r0,0,z0,v0*np.sin(alpha),0,v0*np.cos(alpha)]
tmax=160*a/v0
tab_t=np.linspace(0,tmax,1000)
solution=odeint(derivee,ci,tab_t)
tab_r=solution[:,0]
```

```

tab_z=solution[:,2]
plt.figure(figsize=(8,6))
plt.plot(tab_z,tab_r)
plt.xlabel("z(m)")
plt.ylabel("r(m)")
plt.title("Mouvement d'un électron envoyé vers une lentille magnétique placée en z=0")
plt.show()

```



Ici $Z = -0,1$ et $Z' = 0,04$:

```

[11]: fp=1/(1/0.04-1/(-0.1))
print(fp)

```

0.02857142857142857

```

[14]: z0=-60*a
ci=[r0,0,z0,v0*np.sin(alpha),0,v0*np.cos(alpha)]
tmax=120*a/v0
tab_t=np.linspace(0,tmax,1000)
solution=odeint(derivee,ci,tab_t)
tab_r=solution[:,0]

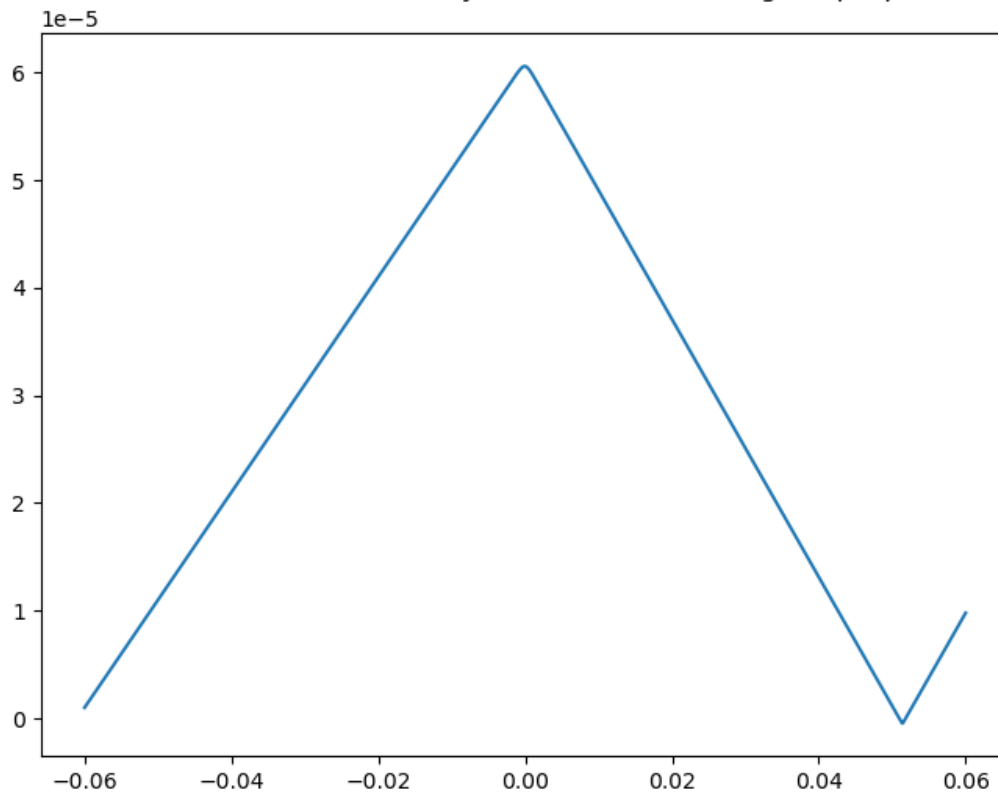
```

```

tab_z=solution[:,2]
plt.figure(figsize=(8,6))
plt.plot(tab_z,tab_r)
plt.title("Mouvement d'un autre électron envoyé vers une lentille magnétique_
↳placée en z=0")
plt.show()

```

Mouvement d'un autre électron envoyé vers une lentille magnétique placée en z=0



```

[15]: fp=1/(1/0.055-1/(-0.06))
print(fp)

```

0.028695652173913042

```

[16]: z0=-100*a
tmax=200*a/v0
tab_t=np.linspace(0,tmax,1000)
plt.figure(figsize=(8,6))
tab_alpha=np.linspace(0.001,0.04,10)
for alpha in tab_alpha:
    ci=[r0,0,z0,v0*np.sin(alpha),0,v0*np.cos(alpha)]
    solution=odeint(derivee,ci,tab_t)
    tab_r=solution[:,0]

```

```

tab_z=solution[:,2]
plt.plot(tab_z,tab_r)
plt.ylim([0,0.004])
plt.title("Mouvement d'électrons lancés depuis le même point avec des angles_
↳différents")
plt.show()

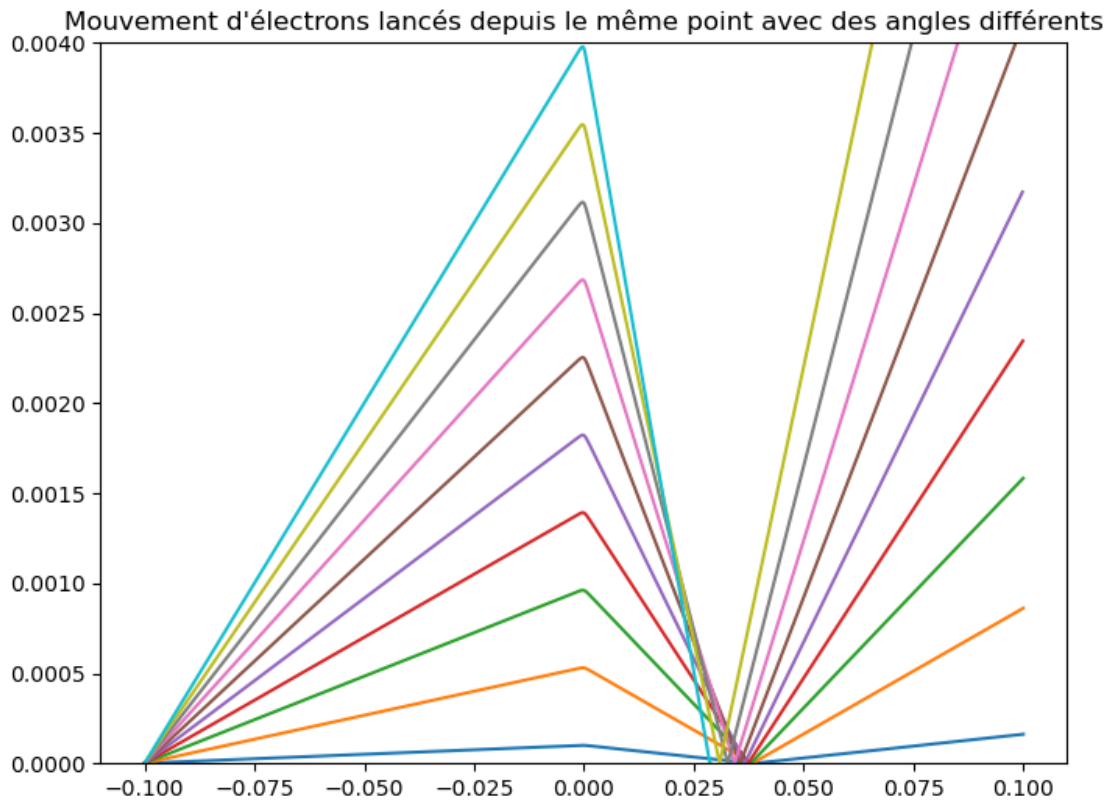
```

/tmp/ipykernel_10499/3324973610.py:8: ODEintWarning: Excess work done on this call (perhaps wrong Dfun type). Run with full_output = 1 to get quantitative information.

```

solution=odeint(derivee,ci,tab_t)

```



On voit apparaître l'aberration sphérique: les rayons les plus inclinés convergent plus tôt.

```

[22]: z0=-100*a
alpha=0.001
tmax=200*a/v0
tab_t=np.linspace(0,tmax,1000)
plt.figure(figsize=(8,6))
tab_B0=np.linspace(25e-3,100e-3,10)
for B0 in tab_B0:
    ci=[r0,0,z0,v0*np.sin(alpha),0,v0*np.cos(alpha)]

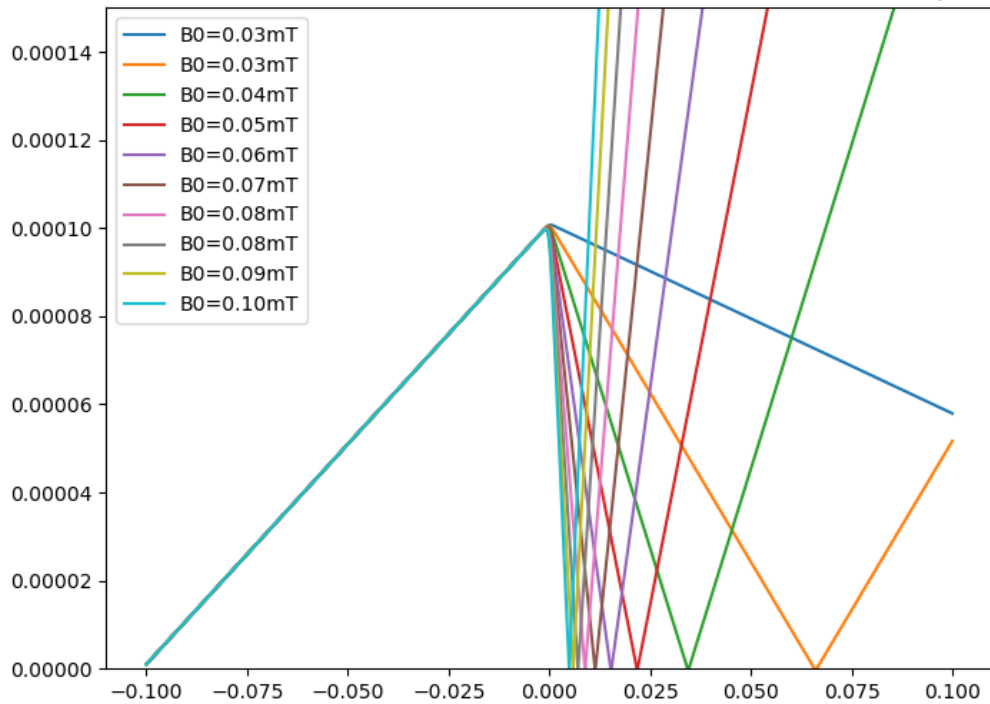
```

```

solution=odeint(derivee,ci,tab_t)
tab_r=solution[:,0]
tab_z=solution[:,2]
plt.plot(tab_z,tab_r,label="B0=%.2fmT"%B0)
plt.legend()
plt.ylim([0,1.5e-4])
plt.title("Mouvements d'électrons lancés avec les mêmes CI dans des lentilles_
↳de champ B différents")
plt.show()

```

Mouvements d'électrons lancés avec les mêmes CI dans des lentilles de champ B différents



```
[23]: print(32*m**2*v0**2/(3*np.pi*a*e**2*(40e-3)**2))
```

0.027457543411333028

```
[ ]:
```