

```

/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 """
4 On trace les lignes de champs correspondant à une spire placée dans le plan
   équatorial
5 en résolvant le système d'équations
6  $dx/dt=B_x(x,z)$ 
7  $dz/dt=B_z(x,z)$ 
8 où  $B_x$  et  $B_z$  sont les composantes du champ magnétique calculée ) partir des
   formules de
9 Biot et Savart
10 """
11 from __future__ import division
12 from scipy import *
13 from pylab import *
14 from scipy.integrate import quad, odeint
15
16 def F(Y, t):
17     [x, z] = Y
18     def dBx(t):
19         r3 = (x**2+a**2+z**2-2*a*x*cos(t))**(1.5)
20         return z*cos(t)/r3
21     Jx, err = quad(dBx, 0, pi)
22     def dBz(t):
23         r3 = (x**2+a**2+z**2-2*a*x*cos(t))**(1.5)
24         return (a-x*cos(t))/r3
25     Jz, err = quad(dBz, 0, pi)
26     return [Jx, Jz]
27
28 #initialisation
29 uini = 0
30 ufin = 1
31 Npas = 500
32 a = 1
33 for xini in arange(-0.8, 0.9, 0.1):
34     print "xini =", xini
35     zini = 0
36     cond_ini = [xini, zini]
37     # intégration vers l'avant
38     u = linspace(uini, ufin, Npas)
39     Yn = odeint(F, cond_ini, u)
40     [x, z] = Yn.T
41     plot(x, z, color='b')
42     # intégration vers l'arrière
43     u = linspace(uini, -ufin, Npas)
44     Yn = odeint(F, cond_ini, u)
45     [x, z] = Yn.T
46     plot(x, z, color='b')

```

```

xini = -0.8
xini = -0.7
xini = -0.6
xini = -0.5
xini = -0.4
xini = -0.3
xini = -0.2

```

```

xini = -0.1
xini = -2.22044604925e-16
xini = 0.1
xini = 0.2
xini = 0.3
xini = 0.4
xini = 0.5
xini = 0.6
xini = 0.7
xini = 0.8

```

```

49
50 plot(a, 0, 'or')
51 plot(-a, 0, 'og')
52 xlabel(ur"x(t)")
53 ylabel(ur"z(t)")
54 axis('equal')
55
56 show()

```

