```
# -*- coding: utf-8 -*-
import numpy
import random
import math
from matplotlib.pyplot import *
import isingchampvrai2

N = 30
c=numpy.zeros((N,N),dtype=numpy.uint8)
for k in range(0,N):
    for i in range(0,N):
        c[k,i]=random.randint(0,1)
ising1 = isingchampvrai2.Ising2D(N,c)

'''cette fonction permet de calculer les valeur de M en fonction de H'''
'''on commence par calculer les valeur pour H de 0 a 1(premiere boucle)'''
'''puis les valeur de H allant de 1 a 0 et 0 a -1 (deuxieme et troisieme boucle)'''
'''enfin on calcul M pour H allant de -1 a 0 et 0 a 1 (deux derniere boucles)'''

def fox(k,T):
    M1=numpy.ones(k)
    M2=numpy.ones(k)
    M3=numpy.ones(k)
    M4=numpy.ones(k)
    M5=numpy.ones(k)
    a=numpy.ones((30,30),dtype=numpy.uint8)
    global d
    for i in range(0,k):
        if i==0:
            h=0.2*i
            ising1.temperature(T,h)
            (m,M,dM,b)=ising1.boucle(1500)
            M1[i]=M
            print(M1[i])
            mask = numpy.ones((30,30),dtype=numpy.uint8)
            b=numpy.bitwise_and(b,mask)
            ising2 = isingchampvrai2.Ising2D(N,b)
        else:
            h=0.2*i
            ising2.temperature(T,h)
            (m,M,dM,g)=ising2.boucle(1500)
            mask = numpy.ones((30,30),dtype=numpy.uint8)
            g=numpy.bitwise_and(g,mask)
            M1[i]=M
            print(M1[i])
            ising2 = isingchampvrai2.Ising2D(N,g)
            a=g
            d=ising2
    for i in range(0,k):
        if i==0:
            h=1-0.2*i
            d.temperature(T,h)
```

```
        (m,M,dM,b)=d.boucle(1500)
        M2[i]=M
        print(M2[i])
        mask = numpy.ones((30,30),dtype=numpy.uint8)
        b=numpy.bitwise_and(a,mask)
        ising2 = isingchampvrai2.Ising2D(N,b)
    else:
        h=1-0.2*i
        ising2.temperature(T,h)
        (m,M,dM,g)=ising2.boucle(1500)
        mask = numpy.ones((30,30),dtype=numpy.uint8)
        g=numpy.bitwise_and(g,mask)
        M2[i]=M
        print(M2[i])
        ising2 = isingchampvrai2.Ising2D(N,g)
        a=g
        d=ising2
for i in range(0,k):
    if i==0:
        h=-0.2*i
        d.temperature(T,h)
        (m,M,dM,b)=d.boucle(1500)
        M3[i]=M
        print(M3[i])
        mask = numpy.ones((30,30),dtype=numpy.uint8)
        b=numpy.bitwise_and(a,mask)
        ising2 = isingchampvrai2.Ising2D(N,b)
    else:
        h=-0.2*i
        ising2.temperature(T,h)
        (m,M,dM,g)=ising2.boucle(1500)
        mask = numpy.ones((30,30),dtype=numpy.uint8)
        g=numpy.bitwise_and(g,mask)
        M3[i]=M
        print(M3[i])
        ising2 = isingchampvrai2.Ising2D(N,g)
        a=g
        d=ising2
for i in range(0,k):
    if i==0:
        h=-1+0.2*i
        d.temperature(T,h)
        (m,M,dM,b)=d.boucle(1500)
        M4[i]=M
        print(M4[i])
        mask = numpy.ones((30,30),dtype=numpy.uint8)
        b=numpy.bitwise_and(a,mask)
        ising2 = isingchampvrai2.Ising2D(N,b)
    else:
        h=-1+0.2*i
        ising2.temperature(T,h)
        (m,M,dM,g)=ising2.boucle(1500)
```

```
            mask = numpy.ones((30,30),dtype=numpy.uint8)
            g=numpy.bitwise_and(g,mask)
            M4[i]=M
            print(M4[i])
            ising2 = isingchampvrai2.Ising2D(N,g)
            a=g
            d=ising2
    for i in range(0,k):
        if i==0:
            h=0.2*i
            d.temperature(T,h)
            (m,M,dM,b)=d.boucle(1500)
            M5[i]=M
            print(M5[i])
            mask = numpy.ones((30,30),dtype=numpy.uint8)
            b=numpy.bitwise_and(a,mask)
            ising2 = isingchampvrai2.Ising2D(N,b)
        else:
            h=0.2*i
            ising2.temperature(T,h)
            (m,M,dM,g)=ising2.boucle(1500)
            mask = numpy.ones((30,30),dtype=numpy.uint8)
            g=numpy.bitwise_and(g,mask)
            M5[i]=M
            print(M5[i])
            ising2 = isingchampvrai2.Ising2D(N,g)
    return M1,M2,M3,M4,M5

M1,M2,M3,M4,M5=fox(6,2.3)

'''les jeux de valeurs de la courbe publie'''
'''voici le resultat de la premiere boucle (ne figure pas sur le graphe pour eviter de surcharger'''

l1=[-0.284843333333,
0.315506944444,
0.516823888889,
0.73861,
0.642823055556,
0.626445555556]

'''les valeur de M pour H allant de 1 a -1'''

l2=[0.800034166667,
0.776038333333,
0.609663611111,
0.648822777778,
0.334531111111,
-0.0289958333333,
-0.147974166667,
-0.725733611111,
-0.872888333333,
-0.917558611111,
```

-0.940065555556,
-0.955733611111]

'''les valeur de M pour H allant de -1 a 1'''

l3=[-0.955714166667,
-0.938275833333,
-0.913613333333,
-0.868986111111,
-0.740529722222,
-0.261173055556,
-0.293708055556,
0.319058055556,
0.600570277778,
0.748796944444,
0.794313888889,
0.821201388889]

t2=np.linspace(-1,1,12)
l2=l2.reverse()

'''on remet les valeur de l2 dans un ordre croissant pour un tracer correcte'''
'''on plot'''

xlabel('champ(H)')
ylabel('moment')
title('hystérésis')
plot(t2,l2,'g',t2,l3,'c')