

# 蒙特卡洛模拟-伊辛模型

---

## 代码实现

---

```

import numpy as np
import matplotlib.pyplot as plt
from numba import jit, int8, float32
J = 1                                #初始化参数
kT = 1
ising = np.zeros((7,7))
def init():                           #初始化模型
    for i in range(1,6):
        for j in range(1,6):
            np.random.seed()
            a = np.random.random_sample()
            if a>=0.5:
                ising[i][j]=1
            else:
                ising[i][j]=-1
init()
@jit(float32(int8,int8,int8[:]))
def H(i,j,A=ising):                  #定义能量
    e_ij = -J/2*(A[i][j]*(A[i-1][j]+A[i+1][j]+A[i][j-1]+A[i][j+1]))
    return e_ij
@jit(float32(int8,int8,int8[:]))
def P(i,j,A=ising):                  #计算翻转概率
    delta_h = -4*H(i,j)
    p = np.exp(-delta_h/kT)
    return p
@jit(float32(int8[:]))
def calc_energy(A=ising):             #计算模型能量
    e = 0
    for i in range(1,6):
        e_i = 0
        for j in range(1,6):
            e_ij = H(i,j)
            e_i = e_i + e_ij
        e = e + e_i
    return e
energy_list = np.array([])            #蒙特卡洛模拟
for _ in range(100000):
    if _%100 == 0:
        print(_)
        np.random.seed()
    i = np.random.randint(1,5)
    j = np.random.randint(1,5)
    ran = np.random.random_sample()
    p = P(i,j)
    if ran < p:
        ising[i][j] = -ising[i][j]
    else:
        pass
    e = calc_energy()
    energy_list = np.append(energy_list,e)
energy = np.average(energy_list)      #计算能量平均值
print(energy)

```

由于python的计算速度过慢，后改用go语言实现并编译成可执行文件(输出的是不同温度对应的不同能量的列表)

```

package main
import (
    "math/rand"
    "math"
    "time"
    "fmt"
)
var J = 1.0
var kT float64
var ising =[7][7] int {{0,0,0,0,0,0,0},{0,0,0,0,0,0,0},{0,0,0,0,0,0,0},{0,0,0,0,0,0,0},
{0,0,0,0,0,0,0},{0,0,0,0,0,0,0},{0,0,0,0,0,0,0}}

func H(i, j int)float64{
    e_ij := -J/2*float64(ising[i][j]*(ising[i-1][j]+ising[i+1][j]+ising[i][j-1]+ising[i][j+1]))
    return e_ij
}
func P(i, j int)float64{
    delta_h := -4*H(i,j)
    p := math.Exp(-delta_h/kT)
    return p
}
func calc_energy()float64{
    e := 0.0
    for i:=1;i<6;i++){
        e_i := 0.0
        for j:=1;j<6;j++ {
            e_ij := H(i,j)
            e_i = e_i + e_ij
        }
        e = e + e_i
    }
    return e
}
func random_int(a, b int)int{
    var x int
    for b>0{
        x = rand.Intn(b+1)
        if x>a-1{
            break;
        }
    }
    return x
}
func main(){

var energy_list [100000] float64
var energy,ran,p,sum_energy float64
var m,n int
for i:=1;i<6;i++){
    for j:=1;j<6;j++){
        rand.Seed(int64(time.Now().Nanosecond()))

        a:=rand.Float64()

```

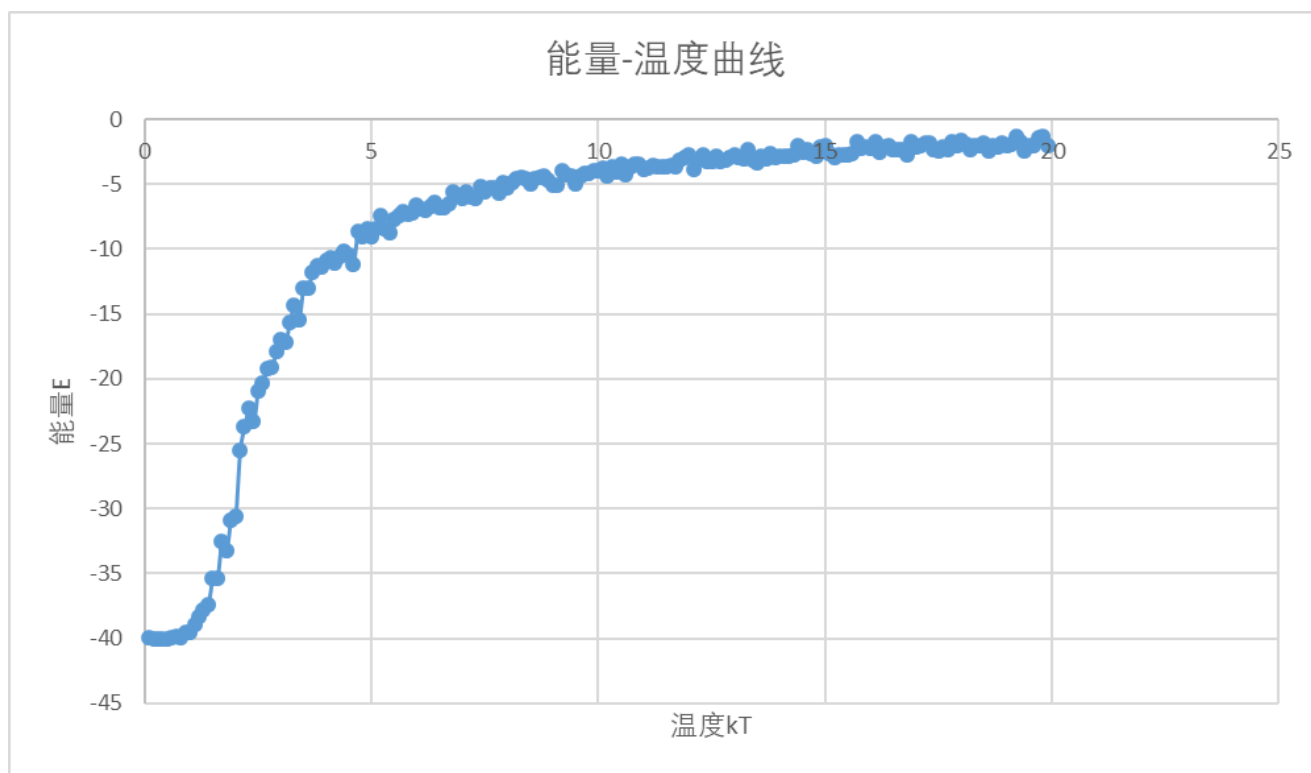
```

        if a>=0.5{
            ising[i][j]=1
        }else{
            ising[i][j]=-1
        }
    }
}
for kT = 0.1;kT<20;kT = kT+0.1{
for k:=0;k<100000;k++){
    if k%100 == 0{
        rand.Seed(int64(time.Now().Nanosecond()))
    }
    m = random_int(1,5)
    n = random_int(1,5)
    ran = rand.Float64()
    p = P(m,n)
    if ran<p{
        ising[m][n] = -ising[m][n]
    }
    energy_list[k] = calc_energy()
}
sum_energy = 0
for l:=0;l<100000;l++){
    sum_energy = sum_energy + energy_list[l]
}
energy = sum_energy/100000
fmt.Println(energy)
}
}

```

## 绘制E-kT图像

---



可观测到在2到3出出现明显的相变潜热