import time

import numba

import itertools as it

from numba import jit, njit,config, threading\_layer, prange, cuda, int64

from numba.typed import List

from sSquareEngineGPU import s2

from setup import readInput

model, nSite, subSpace, nStates, s2Target, maxItr, startSpinTargetItr, energyTola, spinTola, beta, jVal, det, Ms, posibleDet, bondOrder, outputfile, restart, saveBasis = readInput()

sList = []

for i in range(nSite):

sList.append(i)

sProduct = []

sProduct = list(it.product(sList, repeat=2))

typed\_sProduct = List()

[typed\_sProduct.append(x) for x in sProduct]

def spinCalculator(basis, energy, ci, lenBasis, Final):

nbBasis = List()

for i in prange(lenBasis):

nbBasis.append( basis[i].bin )

nbEnergy = List()

[nbEnergy.append(x) for x in energy]

s2List = List()

[s2List.append(0.0) for x in range(nStates)]

for xx in range (nStates):

ciOneState = List()

[ciOneState.append(x) for x in ci[(lenBasis \* xx) :(lenBasis \* (xx+1))]]

s2List[xx] = s2( nSite, nbBasis, ciOneState, typed\_sProduct, lenBasis)

if Final:

with open(outputfile,"a") as fout:

newline = ("\nEnergy & Spin Value of First %d States.\n")%(nStates)

fout.write(newline)

for xx in range (nStates):

newline = ("State: %d\tEnergy: %f\ts^2 Expe Val: %2.4f\n")%( (xx + 1), round(nbEnergy[xx],6), s2List[xx])

#print("State:",(xx+1),"\tEnergy: ", round(nbEnergy[xx],6),"\t s^2 Expe Val: ", s2List[xx])

with open(outputfile,"a") as fout:

fout.write(newline)

return s2List

def stateFinder( s2ValList, s2Target):

diffList = []

for i in range(nStates):

diff = abs(s2Target - s2ValList[i])

diffList.append(diff)

n = diffList.index(min(diffList))

return n, abs(s2ValList[n] - s2Target)