#!/usr/bin/python

import numpy

from numpy import linalg

from setup import readInput

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

f1 = open(bondOrder)

line1 = f1.readline()

bO1 = []

bO2 = []

while line1:

values = line1.split()

bO1.append(int(values[0])-1)

bO2.append(int(values[1])-1)

line1 = f1.readline()

f1.close()

j2BondOrder = bondOrder+"2"

f2 = open(j2BondOrder)

line2 = f2.readline()

bO3 = []

bO4 = []

while line2:

values = line2.split()

bO3.append(int(values[0])-1)

bO4.append(int(values[1])-1)

line2 = f2.readline()

f2.close()

cSz = -jVal \* 0.25

cSxSy = -jVal \* 0.50

jRatio = 0.5 #(j2 / j1)

def subSited(a1, a2):

diff = 0

for i in range(nSite):

if a1[i] != a2[i]:

diff += 1

return diff

def opSz(a):

Sz = 0.0

for i, x in enumerate(bO1):

if (a[bO1[ i]] == a[ bO2[ i]]):

Sz += cSz

else:

Sz -= cSz

for i, x in enumerate(bO3):

if (a[bO3[ i]] == a[ bO4[ i]]):

Sz += (cSz) \* jRatio

else:

Sz -= cSz \* jRatio

return Sz

def opSxSy(a, b):

SxSy = 0.0

for i, x in enumerate(bO1):

if ((a[bO1[i]] != b[bO1[i]] and a[bO2[i]] != b[bO2[i]]) and a[bO1[i]] != a[bO2[i]] ):

SxSy += cSxSy

for i, x in enumerate(bO3):

if ((a[bO3[i]] != b[bO3[i]] and a[bO4[i]] != b[bO4[i]]) and a[bO3[i]] != a[bO4[i]] ):

SxSy += (cSxSy) \* jRatio

return SxSy

def Hamiltonian(A):

lenA = len(A)

Hsub = numpy.zeros((lenA,lenA))

for idx, x in enumerate(A) :

for idy, y in enumerate(A) :

siteDiff = subSited(x, y)

if siteDiff == 0:

Hsub[idx][idy] = opSz(x.bin)

if siteDiff == 2:

Hsub[idx][idy] = opSxSy(x.bin, y.bin)

return Hsub