Intento de construcción del problema de los mubs en un model de Iing, queremos algo con la forma

$$H = \sum_{j,k} J_{j,k} S_j S_k$$

Donde $J_{j,k}$ puede tener tanto valore positivos como negativos o cero para que las configuraciones de mínima energía sean Mubs

Considerando que las entradas de las matrices tienen la forma

```
exp(%i*2*%pi*l/L)/sqrt(d)
```

donde l = 0:1:L-1, siendo L el numero de fases diferentes a evaluar y d es la dimensión de la matriz

Entonces por ejemplo con 2 fases diferentes en d=2 necesitamos 4 qbits para representar un vector.

Con L fases necesitamos d^L qbits para representar un vector de dimensión

Para 2 fases no hay mubs pero podemos intentar usar esta configuracion para tratar de entender que quermos conseugir

Con d=2 L=2 un vector necesita 4 qb para representar todas las posibles configuraciones de fases

La cadena s1 s2 s3 s4 s5 s6 s7 s8 podemos representar una matriz de 2x2 donde

s1 s2 s3 s4 es el primer vector
s5 s6 s7 s8 es el segundo vector

con las dos fases a y b (ab es a en la primer entrada b en la segunda)

s:	1	2	3	4
aa	-1	0	0	0
ab	0	-1	0	0
ba	0	0	-1	0
bb	0	0	0	-1

Como buscamos un modelo de Ising si en cada configuración cambiamos los 0 por 1 y los 1 por 0 representarian los mismo.

El problema es cuando hay más de dos estados up, de momento no hay interpretación para esto.

Para más fases, más dimensiones es igual

Así que lo primero a plantearse es cómo son las interacciones para que la configuración de mínima energía

para la cadena s1 s2 s3 s4 si queremos s1= 1 y todos los demás en cero así que parala fijamos J_{i,i}= a

J_{1,2} = b   J_{1,3} = b   J_{1,4} = b

s:	1	2	3	4
1	a	b	b	b
2	0	a	0	0
3	0	0	a	0
4	0	0	0	a

Con esto el estado de mínima eneríga es
1 0 0 0   y 0 1 1 1

De igual forma podríamos con -b (menos la diagonal) la fila dos en lugar la de 1

podemos cambiar este -b a 1,2

s:	1	2	3	4
1	a	0	0	0
2	b	a	b	b
3	0	0	a	0
4	0	0	0	a

Con esto el estado de mínima eneríga es
0 1 0 0   y 1 0 1 1

Al final poemos intentar algo con la forma

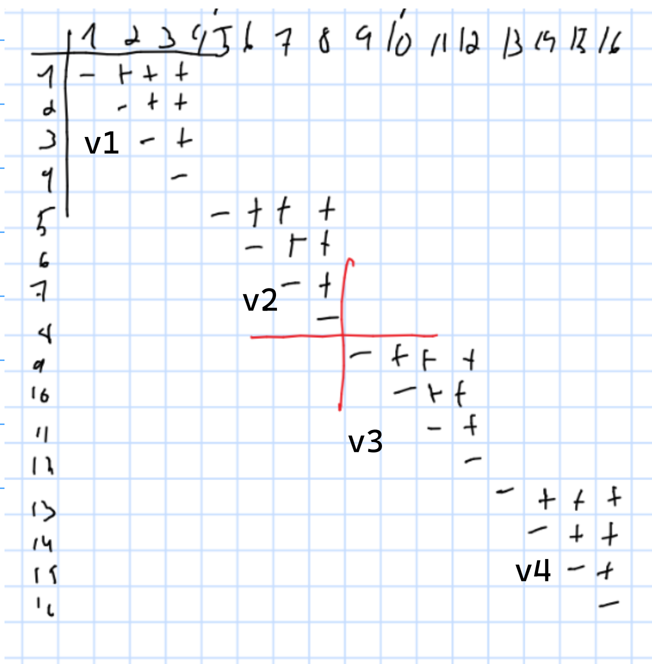
s:	1	2	3	4
aa	1	0	0	0
ab	0	1	0	0
ba	0	0	1	0
bb	0	0	0	1

s:	1	2	3	4
1	a	b	b	b
2	0	a	b	b
3	0	0	a	b
4	0	0	0	a

Para tener estados

Para un set completo en d=2 necesitamos 2 matrices (una ya es la identidad) y asi que son 4 vectores 16 qbit, y esperamos algo de esta forma

mat 1



mat2

Vemos que podemos pensar en cosas que ocurran por bloques

Entonces podemos volvera pensar que pasa solo para un vector

Para d = 2 con solo un vector y L =2 (dos fases) queremos

$$
J_{i,j}= \begin{array}{c|cccc} s: & 1 & 2 & 3 & 4 \\ \hline 1 & a & b & b & b \\ 2 & 0 & a & b & b \\ 3 & 0 & 0 & a & b \\ 4 & 0 & 0 & 0 & a \end{array}
$$

Aquí b contiene el sigo menos

Los casos en rojo son los que queremos que sean mínimos

Pero estos son los mínimos del sistema

					n_+	n_-	$n_+ - n_-$	$4a + \frac{1}{2}(\delta^2-4)$
0	-1	-1	-1	-1	0	4	-4	4a + 6b
1	-1	-1	-1	1	1	3	-	4a
2	-1	-1	1	-1	1	3	-	4a +
3	-1	-1	1	1	2	2	0	4a - 2b
4	-1	1	-1	-1	1	3	-2	4a + 0
5	-1	1	-1	1	2	2	0	4a - 2b
6	-1	1	1	-1	2	2	0	4a - 2b
7	-1	1	1	1	3	1	2	4a + 0
8	1	-1	-1	-1	1	3	-2	4a + 0
9	1	-1	-1	1	2	2	0	4a - 2b
10	1	-1	1	-1	2	2	0	4a - 2b
11	1	-1	1	1	3	1	2	4a + 0
12	1	1	-1	-1	2	2	0	4a - 2b
13	1	1	-1	1	3	1	2	4a + 0
14	1	1	1	-1	3	1	2	4a + 0
15	1	1	1	1	4	0	4	4a + 6b

Las configuraciones de mínima energía son aquellas que tienen la mitad de los qbits en un estado 1

Para 5 qbits tendriamos 2 en el estado 1 y   3 en estado 0, así como todos los casos con 2 en 0 y 3 en 1.

```
E,H,t=findM(2,1,2,0,1);
H
4Ã0097 4 Matrix{Float64}:
 0.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0
 0.0  0.0  0.0  0.0
t
6-element Vector{Int32}:
  4
  6
  7
 10
 11
 13
VecConf.(t.-1,4)
6-element Vector{Vector{Int32}}:
 [1, 1, -1, -1]
 [1, -1, 1, -1]
 [-1, 1, 1, -1]
 [1, -1, -1, 1]
 [-1, 1, -1, 1]
 [-1, -1, 1, 1]
```

```
8Ã0097 8 Matrix{Float64}:
 0.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0
 0.0  0.0  1.0  1.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  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  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

Este tipo de interacción hace que las conf. con mag cercana a cero sean las conf. de minima energía

```
VecConf.(t.-1,8)
36-element Vector{Vector{Int32}}:
 [1, 1, -1, -1, 1, 1, -1, -1]
 [1, -1, 1, -1, 1, 1, -1, -1]
 [-1, 1, 1, -1, 1, 1, -1, -1]
 [1, -1, -1, 1, 1, 1, -1, -1]
 [-1, 1, -1, 1, 1, 1, -1, -1]
 [-1, -1, 1, 1, 1, 1, -1, -1]
 [1, 1, -1, -1, 1, -1, 1, -1]
 [1, -1, 1, -1, 1, -1, 1, -1]
 [-1, 1, 1, -1, 1, -1, 1, -1]
 [1, -1, -1, 1, 1, -1, 1, -1]
 [-1, 1, -1, 1, 1, -1, 1, -1]
 [-1, -1, 1, 1, 1, -1, 1, -1]
 [1, 1, -1, -1, -1, 1, 1, -1]
 [1, -1, 1, -1, -1, 1, 1, -1]
 [-1, 1, 1, -1, -1, 1, 1, -1]
 [1, -1, -1, 1, -1, 1, 1, -1]
 [-1, 1, -1, 1, -1, 1, 1, -1]
 [-1, -1, 1, 1, -1, 1, 1, -1]
 [1, 1, -1, -1, 1, -1, -1, 1]
 [1, -1, 1, -1, 1, -1, -1, 1]
 [-1, 1, 1, -1, 1, -1, -1, 1]
 [1, -1, -1, 1, 1, -1, -1, 1]
 [-1, 1, -1, 1, 1, -1, -1, 1]
 [-1, -1, 1, 1, 1, -1, -1, 1]
 [1, 1, -1, -1, -1, 1, -1, 1]
 [1, -1, 1, -1, -1, 1, -1, 1]
 [-1, 1, 1, -1, -1, 1, -1, 1]
 [1, -1, -1, 1, -1, 1, -1, 1]
 [-1, 1, -1, 1, -1, 1, -1, 1]
 [-1, -1, 1, 1, -1, 1, -1, 1]
 [1, 1, -1, -1, -1, -1, 1, 1]
 [1, -1, 1, -1, -1, -1, 1, 1]
 [-1, 1, 1, -1, -1, -1, 1, 1]
 [1, -1, -1, 1, -1, -1, 1, 1]
 [-1, 1, -1, 1, -1, -1, 1, 1]
```

Aun tenemos que evaluar Ortogonalidad y Mubsness

Creo que esto lo podemos pensensar con las secciones de los qbits que representan algun vector por ejemplo para d=2 k =2

s_{js}

j-1	0
j	1
j+1	0
j+2	0

$$\longrightarrow v = \begin{vmatrix} c*\exp(i*phj\_1) \\ c*\exp(i*phj\_2) \end{vmatrix}$$

s_{ks}

k-2	0
k-1	0
k	1
k+1	0

$$\longrightarrow w = \begin{vmatrix} c*\exp(i*phk\_1) \\ c*\exp(i*phk\_2) \end{vmatrix}$$

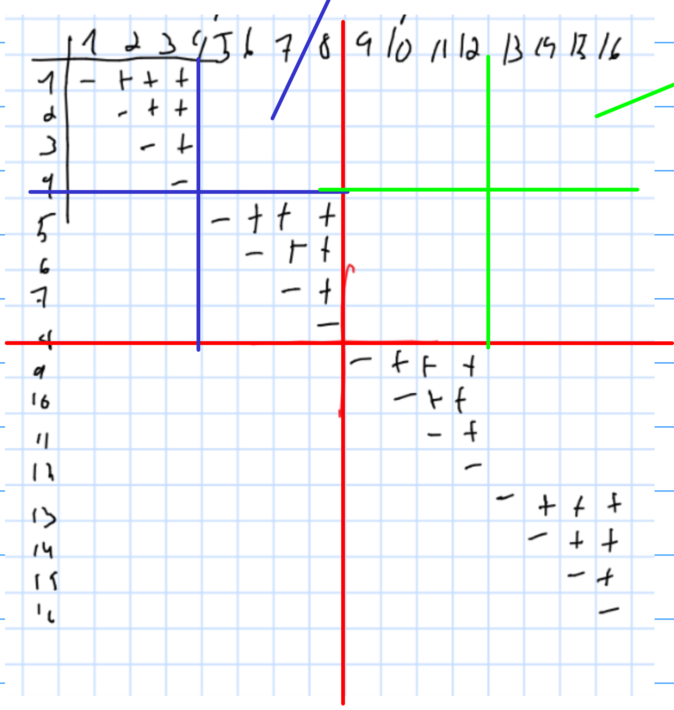
Entonces si v \dot w cumple alguna propiedad deseada premiamos esta configuracción

	k-2	k-1	k	k+1
j-1	a	a	b	a
j	b	b	a	b
j+1	a	a	b	a
j+2	a	a	b	a

por ejemplo
a $\leq$ 0 para que esten en estados iguales
b > 0 para que esten en estados opuestos

Aquí podemos poner interacciones que premiem ortogonalidad

Y aquí pondremos interacciones que premiem Mubsness



Entonces revisamos los pares y premiamos aquellos que sean ortogonales sumamos y construimos J_{i,j}

```
H = Ortogonalidadold(2,2,-1,1)
4Ã[0097]4 Matrix{Float64}:
  0.0  -2.0  -3.0   4.0
 -2.0   0.0   0.0   0.0
 -2.0   0.0   0.0  -1.0
 -4.0  -2.0  -2.0   0.0
Hf = kron([0 1;1 0],H)
8Ã[0097]8 Matrix{Float64}:
  0.0  -0.0  -0.0   0.0   0.0  -2.0  -3.0   4.0
 -0.0   0.0   0.0   0.0  -2.0   0.0   0.0   0.0
 -0.0   0.0   0.0  -0.0  -2.0   0.0   0.0  -1.0
 -0.0  -0.0  -0.0   0.0  -4.0  -2.0  -2.0   0.0
  0.0  -2.0  -3.0   4.0   0.0  -0.0  -0.0   0.0
 -2.0   0.0   0.0   0.0  -0.0   0.0   0.0   0.0
 -2.0   0.0   0.0  -1.0  -0.0   0.0   0.0  -0.0
 -4.0  -2.0  -2.0   0.0  -0.0  -0.0  -0.0   0.0
 E = Energias(H);
    #@show H
    Em = minimum(E);
    minst = findall(isone,abs.(E·-Em).<1e-12);
256-element Vector{Float64}:
```

```
4-element Vector{Int32}:
   1
 106
 151
 256
VecConf.(minst·-1,8)
4-element Vector{Vector{Int32}}:
 [-1, -1, -1, -1, -1, -1, -1, -1]
 [1, -1, -1, 1, -1, 1, 1, -1]
 [-1, 1, 1, -1, 1, -1, -1, 1]
 [1, 1, 1, 1, 1, 1, 1, 1]
```

```
H = Ortogonalidadold(2,2,-1,0.5)
4Ã[0097]4 Matrix{Float64}:
 -1.0  -2.5  -3.0   2.0
 -2.5  -1.0  -1.0  -0.5
 -2.5  -1.0  -1.0  -1.0
 -4.0  -2.5  -2.5  -1.0
Hf = kron([0 1;1 0],H)
8Ã[0097]8 Matrix{Float64}:
 -0.0  -0.0  -0.0   0.0  -1.0  -2.5  -3.0   2.0
 -0.0  -0.0  -0.0  -0.0  -2.5  -1.0  -1.0  -0.5
 -0.0  -0.0  -0.0  -0.0  -2.5  -1.0  -1.0  -1.0
 -0.0  -0.0  -0.0  -0.0  -4.0  -2.5  -2.5  -1.0
 -1.0  -2.5  -3.0   2.0  -0.0  -0.0  -0.0   0.0
 -2.5  -1.0  -1.0  -0.5  -0.0  -0.0  -0.0  -0.0
 -2.5  -1.0  -1.0  -1.0  -0.0  -0.0  -0.0  -0.0
 -4.0  -2.5  -2.5  -1.0  -0.0  -0.0  -0.0  -0.0
 E = Energias(Hf);
    #@show H
    Em = minimum(E);
    minst = findall(isone,abs.(E·-Em).<1e-12);
```

```
2-element Vector{Int32}:
   1
 256
VecConf.(minst·-1,8)
2-element Vector{Vector{Int32}}:
 [-1, -1, -1, -1, -1, -1, -1, -1]
 [1, 1, 1, 1, 1, 1, 1, 1]
```

Algo más simple parece mejor

Aquí es más facil premiar (usando a)
y casticar (usando b)

	k-2	k-1	k	k+1
j-1	0	0	0	0
j	0	0	a	0
j+1	0	0	0	0
j+2	0	0	0	0

```
H = Ortogonalidad(2,2,-1,0.5)
4Ã4 Matrix{Float64}:
 1.0  -1.0  -1.0   1.0
 0.0   1.0   1.0  -1.0
 0.0   0.0   1.0  -1.0
 0.0   0.0   0.0   1.0
Hf = kron([0 1;1 0],H)
8Ã8 Matrix{Float64}:
 0.0  -0.0  -0.0   0.0  1.0  -1.0  -1.0   1.0
 0.0   0.0   0.0  -0.0  0.0   1.0   1.0  -1.0
 0.0   0.0   0.0  -0.0  0.0   0.0   1.0  -1.0
 0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0
 1.0  -1.0  -1.0   1.0  0.0  -0.0  -0.0   0.0
 0.0   1.0   1.0  -1.0  0.0   0.0   0.0  -0.0
 0.0   0.0   1.0  -1.0  0.0   0.0   0.0  -0.0
 0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0
VecConf.(minst⋅−1,8)
2-element Vector{Vector{Int32}}:
 [1, -1, -1, 1, -1, 1, 1, -1]
 [-1, 1, 1, -1, 1, -1, -1, 1]
```

Para el Mubsness se plantea algo parecido

```
H = Mubsness(2,2,-1,1)
4Ã4 Matrix{Float64}:
 1.0  1.0  1.0  1.0
 0.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0
```

Ni para 2 fases   ni para 3 fases hay mubs

```
H = Mubsness(2,3,-1,1)
9Ã9 Matrix{Float64}:
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
```

```
H = Mubsness(2,4,-1,1)
16Ã16 Matrix{Float64}:
 1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   0.0   0.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   0.0   0.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   0.0   0.0   0.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   0.0   0.0   0.0   0.0   0.0   1.0  -1.0  -1.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0  -1.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0  -1.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0  -1.0  -1.0   1.0  -1.0   1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0   1.0  -1.0   1.0  -1.0
 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0  -1.0   1.0  -1.0
 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   1.0  -1.0   1.0
 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   1.0  -1.0
 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   1.0
```

Hasta 4 sabemos que hay

```
H = Mubsness(2,2,-1,1)
4Ã4 Matrix{Float64}:
 1.0  1.0  1.0  1.0
 0.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0
H1 = kron(ones(2,2),H)
8Ã8 Matrix{Float64}:
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0
 0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0
H2 = kron(Ia(2),H1)
16Ã16 Matrix{Float64}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  1.0  1.0  1.0  0.0  1.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  1.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
HM= HMubs(2,2,2,-1,1)
16Ã16 Matrix{Float64}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  1.0  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  1.0  1.0  1.0  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  1.0  1.0  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  1.0  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  1.0  1.0  1.0  1.0
 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  1.0  1.0  1.0
 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  1.0  1.0
 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  1.0
 1.0  1.0  1.0  1.0  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  1.0  1.0  1.0  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  1.0  1.0  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  1.0  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  1.0  1.0  1.0  1.0  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  1.0  1.0  1.0  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  1.0  1.0  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  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
HO = HOrth(2,2,2,-1,1)
16Ã16 Matrix{Float64}:
 0.0  -0.0  -0.0   0.0  1.0  -1.0  -1.0   1.0  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   1.0   1.0  -1.0  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   1.0  -1.0  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   1.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   0.0
 1.0  -1.0  -1.0   1.0  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   1.0   1.0  -1.0  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   1.0  -1.0  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   1.0  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  1.0  -1.0  -1.0   1.0
 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   1.0   1.0  -1.0
 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   1.0  -1.0
 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   1.0
 0.0   0.0   0.0   0.0  0.0   0.0   0.0   0.0  1.0  -1.0  -1.0   1.0  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   1.0   1.0  -1.0  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   1.0  -1.0  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   1.0  0.0   0.0   0.0   0.0
H1up = oneUpforVec(2,2,2,0,1)
16Ã16 Matrix{Float64}:
 0.0  1.0  1.0  1.0  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  1.0  1.0  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  1.0  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  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  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  1.0  1.0  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  1.0  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
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0  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  1.0  1.0  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  1.0  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  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  1.0  1.0
 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  1.0  1.0
 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  1.0
 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
```

```
H = H1up+ HO + HM
16Ã16 Matrix{Float64}:
 0.0   1.0   1.0   1.0  1.0  -1.0  -1.0   1.0  1.0   1.0   1.0   1.0  0.0   0.0   0.0   0.0
 0.0   0.0   1.0   1.0  0.0   1.0   1.0  -1.0  0.0   1.0   1.0   1.0  0.0   0.0   0.0   0.0
 0.0   0.0   0.0   1.0  0.0   0.0   1.0  -1.0  0.0   0.0   1.0   1.0  0.0   0.0   0.0   0.0
 0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0  0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0
 1.0  -1.0  -1.0   1.0  0.0   1.0   1.0   1.0  0.0   0.0   0.0   0.0  1.0   1.0   1.0   1.0
 0.0   1.0   1.0  -1.0  0.0   0.0   1.0   1.0  0.0   0.0   0.0   0.0  0.0   1.0   1.0   1.0
 0.0   0.0   1.0  -1.0  0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0  0.0   0.0   1.0   1.0
 0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0
 1.0   1.0   1.0   1.0  0.0   0.0   0.0   0.0  0.0   1.0   1.0   1.0  1.0  -1.0  -1.0   1.0
 0.0   1.0   1.0   1.0  0.0   0.0   0.0   0.0  0.0   0.0   1.0   1.0  0.0   1.0   1.0  -1.0
 0.0   0.0   1.0   1.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0  0.0   0.0   1.0  -1.0
 0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0
 0.0   0.0   0.0   0.0  1.0   1.0   1.0   1.0  1.0  -1.0  -1.0   1.0  0.0   1.0   1.0   1.0
 0.0   0.0   0.0   0.0  0.0   1.0   1.0   1.0  0.0   1.0   1.0  -1.0  0.0   0.0   1.0   1.0
 0.0   0.0   0.0   0.0  0.0   0.0   1.0   1.0  0.0   0.0   1.0  -1.0  0.0   0.0   0.0   1.0
 0.0   0.0   0.0   0.0  0.0   0.0   0.0   1.0  0.0   0.0   0.0   1.0  0.0   0.0   0.0   0.0
```