

# Υπολογιστική κβαντική φυσική και εφαρμογές

---

ΥΦΥ201

ΣΤΕΦΑΝΙΑ ΑΡΧΟΝΤΗ

AEM: 4396

# Περιεχόμενα

1) Listing 6.22: Entangle

2) Listing 6.21: Hyperfine

## Listing 6.22

Για διεμπλεγμένες κβαντικές καταστάσεις, υπολογίζουμε:

A) Την Χαμιλτονιανή

B) Τις ιδιοτιμές και τα ιδιοδιανύσματα

# Κώδικας

- 1) Εισαγωγή των απαραίτητων βιβλιοθηκών
- 2) Αρχικοποίηση παραμέτρων
- 3) Υπολογισμός των tensor products
- 4) Υπολογισμός της Χαμιλτονιανής του συστήματος

```
from numpy import *
from numpy.linalg import *
import numpy as np

nmax = 4
H = zeros((nmax, nmax), float)

# Pauli matrices in order to calculate tensor products
sigma_x = np.array([[0, 1], [1, 0]])
sigma_y = np.array([[0, -1j], [1j, 0]])
sigma_z = np.array([[1, 0], [0, -1]])
XAXB = np.tensordot(sigma_x, sigma_x, axes=0)
YAYB = np.tensordot(sigma_y, sigma_y, axes=0)
ZAZB = np.tensordot(sigma_z, sigma_z, axes=0)
SASB = XAXB + YAYB + ZAZB - 3 * ZAZB
```

```

# Hamiltonian
print("\nHamiltonian without mu^2/r^3 factor\n", sasb, "\n")

# Eigenvalues and eigenvectors
es, ev = eig(sasb)
print("Eigenvalues \n", np.round(es), "\n")
print("Eigenvectors (in columns)\n", ev, "\n")
# Eigenvectors
phi1 = (ev[0, 0], ev[1, 0], ev[2, 0], ev[3, 0])
phi4 = (ev[0, 1], ev[1, 1], ev[2, 1], ev[3, 1])
phi3 = (ev[0, 2], ev[1, 2], ev[2, 2], ev[3, 2])
phi2 = (ev[0, 3], ev[1, 3], ev[2, 3], ev[3, 3])
# List eigenvectors
basis = [phi1, phi2, phi3, phi4]
# Hamiltonian in new basis
for i in range(0, nmax):
    for j in range(0, nmax):
        term = dot(sasb, basis[i])
        H[i, j] = dot(basis[j], term)
print("Hamiltonian in Eigenvector Basis\n", np.round(H))

```

Κώδικας

Υπολογισμός των ιδιοτιμών  
και των ιδιοδιανυσμάτων

Hamiltonian without  $\mu^2/r^3$  factor

```
[[ -2  0  0  0]
 [  0  2  2  0]
 [  0  2  2  0]
 [  0  0  0 -2]]
```

Eigenvalues

```
[ 4.  0. -2. -2.]
```

Eigenvectors (in columns)

```
[[ 0.          0.          1.          0.         ]
 [ 0.70710678  0.70710678  0.          0.         ]
 [ 0.70710678 -0.70710678  0.          0.         ]
 [ 0.          0.          0.          1.         ]]
```

Hamiltonian in Eigenvector Basis

```
[[ 4.  0.  0.  0.]
 [ 0. -2.  0.  0.]
 [ 0.  0. -2.  0.]
 [ 0.  0.  0.  0.]
```

# Αποτελέσματα

---

## Listing 6.21

Στόχος του προγράμματος είναι να απεικονίσει διαγραμματικά την διάσπαση των θεμελιωδών καταστάσεων του ατόμου του υδρογόνου (singlet + triplet), που προκαλείται από εξωτερικό μαγνητικό πεδίο.

# Κώδικας

1) Εισαγωγή βιβλιοθηκών

2) Ορισμός Χαμιλτονιανών  
(Παρουσίας εξωτερικού μαγνητικού πεδίου  
κατά την διεύθυνση Z και χωρίς)

3) Υπολογισμός Ιδιοτιμών

4) Αντικατάσταση των συμβόλων  $\mu_e$  και  $\mu_p$  με τιμές 1 και 0 αντίστοιχα

```
import numpy as np
import matplotlib.pyplot as plt
from sympy import *

W, mue, mup, B = symbols("W mu_e mu_p B")
H = Matrix([[W, 0, 0, 0], [0, -W, 2*W, 0], [0, 2*W, -W, 0], [0, 0, 0, W]])
Hmag = Matrix([[-(mue+mup)*B, 0, 0, 0], [0, -(mue-mup)*B, 0, 0], [0, 0, -(-mue+mup)*B, 0], [0, 0, 0, (mue+mup)*B]])
print("\nHyperfine Hamiltonian H=", H)
print("\n Eigenvalues and multiplicities of H=", H.eigenvals())
print("\n Hmag=", Hmag)
Htot = H + Hmag
print("\n Htot = H + Hmag = ", Htot)
print("\n Eigenvalues of matrix HB")
e1, e2, e3, e4 = Htot.eigenvals()
print("e1=", e1, "\ne2=", e2, "\ne3=", e3, "\ne4=", e4)
print("\n After substitute mu_e=1, and mu_p=0 in eigenvalues")
print("e1=", e1.subs([(mue, 1), (mup, 0)]), "\ne2=", e2.subs([(mue, 1), (mup, 0)]))
print("e3=", e3.subs([(mue, 1), (mup, 0)]), "\ne4=", e4.subs([(mue, 1), (mup, 0)]))
```



```
b = np.arange(0, 4, 0.1)
E = 1
E4 = -E+np.sqrt(b**2+4*E**2)
E3 = E-b
E2 = E+b
E1 = -E-np.sqrt(b**2+4*E**2)
plt.figure()
plt.plot(b, E1, label="E1")
plt.plot(b, E2, label="E2")
plt.plot(b, E3, label="E3")
plt.plot(b, E4, label="E4")
plt.legend()
plt.text(-0.4, 1, "E")
plt.xlabel("Magnetic Field B")
plt.title("Hyperfine Splitting of H Atom 1S Level")
plt.show()
```

## Κώδικας

- 1) Υπολογισμός ενεργειακών καταστάσεων
- 2) Σχεδιασμός διαγράμματος

# Αποτελέσματα

---

```
Hyperfine Hamiltonian H= Matrix([[W, 0, 0, 0], [0, -W, 2*W, 0], [0, 2*W, -W, 0], [0, 0, 0, W]])
```

```
Eigenvalues and multiplicities of H= {W: 3, -3*W: 1}
```

```
Hmag= Matrix([[B*(-mu_e - mu_p), 0, 0, 0], [0, B*(-mu_e + mu_p), 0, 0], [0, 0, B*(mu_e - mu_p), 0], [0, 0, 0, B*(mu_e + mu_p)]])
```

```
Htot = H + Hmag = Matrix([[B*(-mu_e - mu_p) + W, 0, 0, 0], [0, B*(-mu_e + mu_p) - W, 2*W, 0], [0, 2*W, B*(mu_e - mu_p) - W, 0], [0, 0, 0, B*(mu_e + mu_p) + W]])
```

```
Eigenvalues of matrix HB
```

```
e1= -B*mu_e - B*mu_p + W
```

```
e2= -W - sqrt(B**2*mu_e**2 - 2*B**2*mu_e*mu_p + B**2*mu_p**2 + 4*W**2)
```

```
e3= -W + sqrt(B**2*mu_e**2 - 2*B**2*mu_e*mu_p + B**2*mu_p**2 + 4*W**2)
```

```
e4= B*mu_e + B*mu_p + W
```

```
After substitute mu_e=1, and mu_p=0 in eigenvalues
```

```
e1= -B + W
```

```
e2= -W - sqrt(B**2 + 4*W**2)
```

```
e3= -W + sqrt(B**2 + 4*W**2)
```

```
e4= B + W
```

# Αποτελέσματα

---

