Mecanica Cuantica I
Tarea 3

Sergio Montoya
David Pachon

Contents

Chapter 1

1.1

Para iniciar este punto quizas lo mejor que podriamos hacer es revisar las notas de clase. Con lo que nos encontramos con la siguiente expresión en las notas para la semana 2, 3 y 4 en el capitulo 3.8.2:

$$\psi_{n\ell m} = \sqrt{\left(\frac{2}{na_0}\right)^3 \frac{(n-\ell-1)!}{2n\left((n+\ell)!\right)^3}} \left(\frac{2r}{na_0}\right)^{\ell} e^{-\frac{r}{na_0}} L_{n-\ell-1}^{2\ell+1}\left(\frac{2r}{na_0}\right) Y_{\ell}^{m}(\theta,\phi)$$

Por lo tanto, lo que nos queda es esencialmente reemplazar:

$$\psi_{433} = \sqrt{\left(\frac{2}{4a_0}\right)^3 \frac{(4-3-1)!}{2\cdot 4\left((4+3)!\right)^3}} \left(\frac{2r}{4a_0}\right)^{3} e^{-\frac{r}{4a_0}} L_{4-3-1}^{2\cdot 3+1}\left(\frac{2r}{4a_0}\right) Y_{3}^{3}(\theta,\phi)$$

Vamos entonces termino a termino.

1.1.1

Para la raiz tenemos:

$$\begin{aligned}
\sqrt{\left(\frac{3}{4a_0}\right)^3 \frac{(4-3-1)!}{2\cdot 4\left((4+3)!\right)^3}} &= \sqrt{\left(\frac{2}{4a_0}\right)^3 \frac{(0)!}{2\cdot 4\left((7)!\right)^3}} \\
&= \sqrt{\left(\frac{1}{2a_0}\right)^3 \frac{1}{2\cdot 4\,(5040)^3}} \\
&= \sqrt{\frac{1}{2^3 a_0^3}\frac{1}{2\cdot 4\cdot 128024064000}} \\
&= \frac{1}{(2a_0)^{\frac{3}{2}}}\sqrt{\frac{1}{8\cdot 2\cdot 4\cdot 128024064000}}
\end{aligned}$$

1.1.2

Para el caso del siguiente termino la simplificación es realmente simple:

$$\begin{aligned}
\left(\frac{2r}{4a_0}\right)^3 &= \left(\frac{1r}{2a_0}\right)^3 \\
&= \frac{r^3}{(2a_0)^3}
\end{aligned}$$

Ahora si combinamos esto con el punto anterior tenemos:

$$= \frac{1}{(2a_0)^{\frac{3}{2}}}\sqrt{\frac{1}{8\cdot 2\cdot 4\cdot 128024064000}}\frac{r^3}{(2a_0)^3}$$

Podemos meter esto en sympy para encontrar una expresión. El script quedaria algo asi:

```
import sympy as sp

r = sp.Symbol("r")
a0 = sp.Symbol("a0", positive=True)


def sqrt(n, ell):
    first_factor = (2/(n*a0))**3
    second_factor = sp.factorial(n - ell - 1)/(2*n*(sp.factorial(n + ell))**3)
    return sp.sqrt(first_factor * second_factor)


def sec_2(n, ell):
    first_factor = 2 * r
    second_factor = n * a0
    return (first_factor/second_factor)**ell


def expr(n, ell):
    return sp.simplify(sqrt(n, ell) * sec_2(n, ell))


expr = expr(4, 3)
sp.print_latex(expr)
```

Que nos da como resultado:

$$\frac{\sqrt{35}r^3}{135475200a_0^{\frac{9}{2}}}$$

1.1.3

Esta expresión no tiene simplificación, va a quedar como

$$e^{-\frac{r}{4a_0}}$$

1.1.4

Para el caso del polinomio de Laguerre revisemo poco a poco

$$L_{4-3-1}^{2\cdot 3+1}\left(\frac{2r}{4a_0}\right) = L_0^7\left(\frac{2r}{4a_0}\right)$$

Ahora, para que esto cumpla la formula de rodrigues veamos que

$$L_{q-p}^p = L_0^7 \iff p = 7 \wedge q = 7$$

Por lo tanto podemos usar la formula de Rodriguez

$$L_{q-p}^p(x) = (-1)^p \left(\frac{d}{dx}\right)^p L_q(x)$$

$$L_{q-p}^p(x) = (-1)^p \left(\frac{d}{dx}\right)^p e^x \left(\frac{d}{dx}\right)^q (e^{-x}x^q)$$

Ahora bien, dado que estamos hablando de una derivada septima esto resultaria absurdamente complejo de hacer a mano. Entonces hagamos uso de sympy para que haga el trabajo duro por nosotros:

Si escribimos la formula en sympy nos quedaria:

```
import sympy as sp

x = sp.Symbol("x")


def derivate_n(n, expr, s=x):
    res = expr
    for _ in range(n):
        res = sp.diff(res, s)
    return sp.simplify(res)


def laguerr_polynomi(p, q):
    factor = (-1)**p
    expr = sp.exp(-x) * x**q
    prim_derivate = derivate_n(q, expr)
    sec_derivate = derivate_n(p, sp.exp(x) * prim_derivate)

    return factor * sec_derivate


print(laguerr_polynomi(7, 7))
```

Lo que nos da como resultado:

5040

Note que este resultado no esta en terminos de (x) por lo tanto, por mucho que en este caso $x = \frac{2r}{4a_0}$ no nos importa este valor y todo este termino se reduce a 1

1.1.5

Ahora lo unico que nos queda es

$$Y_3^3(\theta, \phi)$$

Este responde a la formula:

$$Y_\ell^m(\theta, \phi) = \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta)\, e^{im\phi}$$

Donde P_ℓ^m es el polinomio asociado de legendre y se expresa como:

$$P_\ell^m(x) = \frac{(-1)^m}{2^\ell \ell!}\left(1-x^2\right)^{\frac{m}{2}} \frac{d^{\ell+m}}{dx^{\ell+m}}\left[\left(x^2-1\right)^\ell\right]$$

Una vez mas nos encontramos ante un problema dificil de hacer a mano. Por lo tanto, hagamoslo de nuevo en sympy.

para poner esto en codigo simplemente realizamos un par de funciones:

```
import sympy as sp

x = sp.Symbol("x")
theta = sp.Symbol("theta")
phi = sp.Symbol("phi")


def derivate_n(n, expr, s=x):
    res = expr
    for _ in range(n):
        res = sp.diff(res, s)
```

```
    return sp.simplify(res)


def asociado(ell, m, s=x):
    first_factor = (-1)**m/(2**ell * sp.factorial(ell))
    second_factor = (1 - s**2)**(m/2)
    expr = (s**2 - 1)**ell
    tird_factor = derivate_n((ell + m), expr, s=s)
    return first_factor * second_factor * tird_factor


def Yellm(ell, m):
    first_factor_sqrt = (2*ell + 1)/(4*sp.pi)
    second_factor_sqrt = sp.factorial(ell - m)/sp.factorial(ell + m)
    first_factor = sp.sqrt(first_factor_sqrt * second_factor_sqrt)
    second_factor = asociado(ell, m).subs(x, sp.cos(theta))
    third_factor = sp.exp(sp.I * m * phi)
    return sp.simplify(first_factor * second_factor * third_factor)


sp.print_latex(Yellm(3, 3))
```

Lo que nos da como resultado:

$$-\frac{\sqrt{35}\left(\sin^2(\theta)\right)^{1.5} e^{3i\phi}}{8\sqrt{\pi}}$$

Debo confesar que no comfiamos en primera instancia en este resultado pues parecia bastante arbitrario entonces solo por confirmar revise la siguiente tabla de la wikipedia: https://en.wikipedia.org/wiki/Table_of_spherical_harmonics la cual nos daba el mismo resultado pero con un formato distinto. Eso me da tranquilidad en mi resultado y aqui pongo la transformación necesaria para que estos dos queden expresados de la misma forma:

$$\begin{aligned}
-\frac{\sqrt{35}\left(\sin^2(\theta)\right)^{1.5} e^{3i\phi}}{8\sqrt{\pi}} &= -\frac{1}{8}\frac{\sqrt{35}}{\sqrt{\pi}}\left(\sin^2(\theta)\right)^{1.5} e^{3i\phi} \\
&= -\frac{1}{8}\sqrt{\frac{35}{\pi}} sin^{2\cdot 1.5}(\theta) e^{3i\phi} \\
&= -\frac{1}{8}\sqrt{\frac{35}{\pi}} sin^{3}(\theta) e^{3i\phi}
\end{aligned}$$

Con esto queda tal cual como lo que esperabamos. Ahora bien, juntando absolutamente todo lo que hicimos antes esto nos queda como:

$$\psi_{433} = \frac{\sqrt{35}r^3}{135475200a_0^{\frac{9}{2}}} e^{-\frac{r}{4a_0}} 5040 \left(-\frac{1}{8}\sqrt{\frac{35}{\pi}} sin^3(\theta) e^{3i\phi}\right)$$

1.2

Dado el resultado anterior, lo podemos expresar para nosotros como:

$$\psi_{433} = \frac{\sqrt{35}r^3}{135475200a_0^{\frac{9}{2}}} e^{-\frac{r}{4a_0}} 5040 Y_3^3$$

Con esto entonces $\langle r \rangle$ nos quedaria como:

$$\langle r \rangle = \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} \psi_{433}^* r \psi_{433} r^2 \sin\theta d\theta d\phi dr$$

$$\langle r \rangle = \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} \left|\psi_{433}\right|^2 r^3 \sin\theta d\theta d\phi dr$$

$$\langle r \rangle = \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} \left|\psi_{433}\right|^2 r^3 \sin\theta d\theta d\phi dr$$

$$\langle r \rangle = \int_0^{\infty} \left| \frac{\sqrt{35} r^3}{135475200 a_0^{\frac{9}{2}}} e^{-\frac{r}{4a_0}} 5040 \right|^2 r^3 dr \int_0^{2\pi} \int_0^{\pi} \left|Y_3^3\right|^2 \sin\theta d\theta d\phi$$

$$\int_0^{2\pi} \int_0^{\pi} \left|Y_3^3\right|^2 \sin\theta d\theta d\phi = 1$$

$$\langle r \rangle = \int_0^{\infty} \left| \frac{\sqrt{35} r^3}{135475200 a_0^{\frac{9}{2}}} e^{-\frac{r}{4a_0}} 5040 \right|^2 r^3 dr$$

Podemos poner esta integral en sympy con el siguiente script:

```
import sympy as sp

r = sp.Symbol("r")
a0 = sp.Symbol("a0", positive=True)
x = sp.Symbol("x")


def derivate_n(n, expr, s=x):
    res = expr
    for _ in range(n):
        res = sp.diff(res, s)
    return sp.simplify(res)


def laguerr_polynomi(p, q):
    factor = (-1)**p
    expr = sp.exp(-x) * x**q
    prim_derivate = derivate_n(q, expr)
    sec_derivate = derivate_n(p, sp.exp(x) * prim_derivate)

    return factor * sec_derivate


def sqrt(n, ell):
    first_factor = (2/(n*a0))**3
    second_factor = sp.factorial(n - ell - 1)/(2*n*(sp.factorial(n + ell))**3)
    return sp.sqrt(first_factor * second_factor)


def sec_2(n, ell):
    first_factor = 2 * r
    second_factor = n * a0
    return (first_factor/second_factor)** ell


def expr(n, ell):
```

```
    return sp.simplify(sqrt(n, ell) * sec_2(n, ell))


expr = expr(4, 3) * sp.exp(- (r)/(4*a0)) * laguerr_polynomi(7, 7)
mod_2 = expr * sp.conjugate(expr)
integral = sp.integrate(mod_2 * r**3, (r, 0, sp.oo))
sp.print_latex(sp.simplify(integral))
```

que nos da como resultado

$$18a_0$$

1.3

Ahora para este caso dado que nos siguen pidiendo los rangos angulares completos se sigue cumpliendo que:

$$\int_0^{2\pi}\int_0^{\pi}\left|Y_3^3\right|^2\sin\theta d\theta d\phi = 1$$

Por lo tanto este problema queda reducido a:

$$P = \int_{\frac{a_0}{2}}^{a_0}\left|\frac{\sqrt{35}r^3}{135475200a_0^{\frac{9}{2}}}e^{-\frac{r}{4a_0}}5040\right|^2 r^2dr$$

Me saltare el desarrollo de esto pues es esencialmente lo mismo que el de la seccion anterior solo cambiando los limites de la integral y no teniendo un r que es la del operador r.

Ahora con esto entonces metamoslo en sympy de manera esencialmente equivalente al punto anterior con el siguiente script:

```
import sympy as sp

r = sp.Symbol("r")
a0 = sp.Symbol("a0", positive=True)
x = sp.Symbol("x")


def derivate_n(n, expr, s=x):
    res = expr
    for _ in range(n):
        res = sp.diff(res, s)
    return sp.simplify(res)


def laguerr_polynomi(p, q):
    factor = (-1)**p
    expr = sp.exp(-x) * x**q
    prim_derivate = derivate_n(q, expr)
    sec_derivate = derivate_n(p, sp.exp(x) * prim_derivate)

    return factor * sec_derivate


def sqrt(n, ell):
    first_factor = (2/(n*a0))**3
    second_factor = sp.factorial(n - ell - 1)/(2*n*(sp.factorial(n + ell))**3)
    return sp.sqrt(first_factor * second_factor)
```

```
def sec_2(n, ell):
    first_factor = 2 * r
    second_factor = n * a0
    return (first_factor/second_factor)**ell


def expr(n, ell):
    return sp.simplify(sqrt(n, ell) * sec_2(n, ell))


expr = expr(4, 3) * sp.exp(-(r)/(4*a0)) * laguerr_polynomi(7, 7)
mod_2 = expr * sp.conjugate(expr)
integral = sp.simplify(sp.integrate(mod_2 * r**2, (r, a0/2, a0)))
sp.print_latex(integral)
sp.print_latex(integral.evalf())
```

y esto nos da como resultado:

$$P = -\frac{17017969}{10321920e^{\frac{1}{2}}} + \frac{3392923553}{2642411520e^{\frac{1}{4}}}$$
$$P \approx 3.42709384773915 \cdot 10^{-9}$$

Chapter 2

Chapter 3

3.1

En este caso

$$1 = \int_0^{\infty}\int_0^{\pi}\int_0^{2\pi} \left|\psi_{2,1,-1}\right|^2 r^2 \sin\theta d\phi d\theta dr$$
$$1 = \int_0^{\infty}\int_0^{\pi}\int_0^{2\pi} \left|Ne^{-\frac{r}{2a_0}} Y_1^{-1}\left(\theta,\phi\right)\right|^2 r^2 \sin\theta d\phi d\theta dr$$
$$1 = |N|^2 \int_0^{\infty}\int_0^{\pi}\int_0^{2\pi} \left|e^{-\frac{r}{2a_0}} Y_1^{-1}\left(\theta,\phi\right)\right|^2 r^2 \sin\theta d\phi d\theta dr$$
$$1 = |N|^2 \int_0^{\infty} e^{-\frac{r}{2a_0}} r^2 dr \int_0^{\pi}\int_0^{2\pi} \left|Y_1^{-1}\left(\theta,\phi\right)\right|^2 \sin\theta d\phi d\theta$$

Con esto lo podemos partir en factora radial y angular lo que nos permitiria acomodar como:

3.1.1 Factor Radial

$$\int_0^{\infty} \left|e^{-\frac{r}{2a_0}}\right|^2 r^2 dr = \int_0^{\infty} e^{-\frac{r}{a_0}} r^2 dr$$

Esta integral la podemos hacer con sympy como:

```
import sympy as sp

r = sp.Symbol("r", positive=True)
a0 = sp.Symbol("a0", positive=True)

exp = sp.exp(- r / (2 * a0))**2 * r**2
result = sp.integrate(exp, (r, 0, sp.oo))
sp.print_latex(result)
```

lo que nos da como resultado:

$$2a_0^3$$

3.1.2 Factor Angular

En este caso, tambien podemos hacer esta integral con sympy como:

```
import sympy as sp

theta, phi = sp.symbols('theta phi', real=True)

norm_const = sp.sqrt(3/(8*sp.pi))
```

```
Y = - norm_const * sp.exp(-sp.I*phi) * sp.sin(theta)
Y_abs2 = Y * sp.conjugate(Y)
Y_abs2 = sp.simplify(Y_abs2)
expr = Y_abs2 * sp.sin(theta)

integral = sp.integrate(expr, (phi, 0, 2*sp.pi), (theta, 0, sp.pi))

sp.print_latex(integral)
```

Lo que nos da como resultado

$$1$$

Con todo esto podemos rebovinar y encontrar el valor de N. Teniamos:

$$\begin{aligned} 1 &= |N|^2 \int_0^{\infty} e^{-\frac{r}{2a_0}} r^2 dr \int_0^{\pi} \int_0^{2\pi} \left|Y_1^{-1}(\theta,\phi)\right|^2 \sin\theta d\phi d\theta \\ 1 &= |N|^2 \left(2a_0^3\right)(1) \\ |N|^2 &= \frac{1}{2a_0^3} \\ |N| &= \frac{1}{\sqrt{2a_0^3}} \end{aligned}$$

Nos quedamos con N positivo y con eso podemos concluir que:

$$N = \frac{1}{\sqrt{2a_0^3}}$$

3.2

Dado que $|\phi|^2$ es la probabilidad este ejercicio se puede plantear como:

$$P = \int_0^{a_0} N^2 r^2 e^{-\frac{r}{a_0}} \int_0^{\frac{\pi}{3}} \int_0^{\frac{\pi}{4}} Y_1^{-1} d\theta d\phi$$

Tomando en cuenta esto podemos trabajar separando cada una de esas integrales y siendo honesto simplemente alterar ligeramente nuestros scripts anteriores para encontrar los resultados:

3.2.1 Parte Radial

En este caso lo unico que debemos hacer es multiplicar nuestra expresión original por $\frac{1}{2a_0^3}$ y cambiar los limites de integración lo que nos deja con el siguiente script

```
import sympy as sp

r = sp.Symbol("r", positive=True)
a0 = sp.Symbol("a0", positive=True)

exp = 1/(2 * a0**3) * sp.exp(- r / (2 * a0))**2 * r**2
result = sp.integrate(exp, (r, 0, a0))
sp.print_latex(result)
```

lo que nos da como resultado:

$$1 - \frac{5}{2e}$$

3.2.2 Parte Angular

En este caso, la unica diferencia real que tenemos que poner con el script para la sección anterior es cambiar los limites de la integral.

```
import sympy as sp

theta, phi = sp.symbols('theta phi', real=True)

norm_const = sp.sqrt(3/(8*sp.pi))

Y = - norm_const * sp.exp(-sp.I*phi) * sp.sin(theta)
Y_abs2 = Y * sp.conjugate(Y)
Y_abs2 = sp.simplify(Y_abs2)
expr = Y_abs2 * sp.sin(theta)

integral = sp.integrate(expr, (phi, 0, sp.pi/3), (theta, 0, sp.pi/4))

sp.print_latex(integral)
```

lo que nos da como resultado:

$$\frac{1}{12} - \frac{5\sqrt{2}}{96}$$

Ahora, volvemos a tener simplemente la relación:

$$\begin{aligned} P &= \left(1 - \frac{5}{2e}\right)\left(\frac{1}{12} - \frac{5\sqrt{2}}{96}\right) \\ &= \left(\frac{2e-5}{2e}\right)\left(\frac{8-5\sqrt{2}}{96}\right) \\ &= \frac{(2e-5)\left(8-5\sqrt{2}\right)}{192e} \end{aligned}$$

Este resultado se puede hacer a mano (como se ve aqui) pero para comprobar mis resultados lo pasamos por un script super facil de sympy que es el siguiente:

```
import sympy as sp

expr1 = 1 - 5/(2*sp.E)
expr2 = 1/12 - (5*sp.sqrt(2))/96

result = sp.simplify(expr1 * expr2)

sp.print_latex(result)
```

y nos dio como resultado

$$\frac{(-5+2e)\left(8.0-5\sqrt{2}\right)}{192e}$$

Que es el mismo resultado.

3.3

En este caso, debemos tomar en consideración que estos operadores estan definidos en cordenadas cartesianas. Por lo tanto, debemos traducirlos primero a coordenadas polares que son las que nos interesan en este caso:

$$\begin{aligned} x &= r\sin\theta\cos\phi \\ y &= r\sin\theta\sin\phi \end{aligned}$$

Ahora con esto partamos de cada uno de esos casos:

3.3.1 x

En este caso nos interesa la integral:

$$\begin{aligned}\int \psi^*(r,\theta,\phi)\, x\psi(r,\theta,\phi)\, d^3r &= \int_0^\infty \int_0^\pi \int_0^{2\pi} x\left|\psi\right|^2 r^2 \sin\theta d\phi d\theta dr \\ &= \int_0^\infty \int_0^\pi \int_0^{2\pi} r\sin\theta\cos\phi\left|\psi\right|^2 r^2 \sin\theta d\phi d\theta dr \\ &= \int_0^\infty \int_0^\pi \int_0^{2\pi} \left|\psi\right|^2 r^3 \sin^2\theta\cos\phi d\phi d\theta dr \\ &= \int_0^\infty N^2 r^3 e^{-\frac{r}{a_0}} dr \int_0^\pi \int_0^{2\pi} \left|Y_1^{-1}\right|^2 \sin^2\theta\cos\phi d\phi d\theta\end{aligned}$$

Esto lo podemos poner en sympy para encontrar los resultados

Parte Radial

Podemos escribir este script como

```
import sympy as sp

r = sp.Symbol("r", positive=True)
a0 = sp.Symbol("a0", positive=True)

exp = 1/(2 * a0**3) * sp.exp(- r / (2 * a0))**2 * r**3
result = sp.integrate(exp, (r, 0, sp.oo))
sp.print_latex(result)
```

Lo que nos da

$$3a_0$$

Parte Angular

Podemos escribir esta integral en sympy con:

```
import sympy as sp

theta, phi = sp.symbols('theta phi', real=True)

norm_const = sp.sqrt(3/(8*sp.pi))

Y = - norm_const * sp.exp(-sp.I*phi) * sp.sin(theta)
Y_abs2 = Y * sp.conjugate(Y)
Y_abs2 = sp.simplify(Y_abs2)
expr = Y_abs2 * sp.sin(theta)**2 * sp.cos(phi)

integral = sp.integrate(expr, (phi, 0, 2*sp.pi), (theta, 0, sp.pi))

sp.print_latex(integral)
```

lo que nos da como resultado:

$$0$$

Ahora bien, retomando nuestros valores esto es:

$$\begin{aligned}\langle x\rangle &= \int_0^\infty N^2 r^3 e^{-\frac{r}{a_0}} dr \int_0^\pi \int_0^{2\pi} \left|Y_1^{-1}\right|^2 \sin^2\theta\cos\phi d\phi d\theta \\ &= 3a_0(0) \\ &= 0\end{aligned}$$

3.3.2 Y

Para este caso nos interesa la integral:

$$\begin{aligned}\int \psi^*(r,\theta,\phi)\, y\psi(r,\theta,\phi)\, d^3r &= \int_0^\infty \int_0^\pi \int_0^{2\pi} y\left|\psi\right|^2 r^2 \sin\theta d\phi d\theta dr \\ &= \int_0^\infty \int_0^\pi \int_0^{2\pi} r\sin\theta\sin\phi\left|\psi\right|^2 r^2 \sin\theta d\phi d\theta dr \\ &= \int_0^\infty \int_0^\pi \int_0^{2\pi} \left|\psi\right|^2 r^3 \sin^2\theta \sin\phi d\phi d\theta dr \\ &= \int_0^\infty N^2 r^3 e^{-\frac{r}{a_0}} dr \int_0^\pi \int_0^{2\pi} \left|Y_1^{-1}\right|^2 \sin^2\theta \sin\phi d\phi d\theta\end{aligned}$$

Con estas integrales entonces podemos pasarlas por sympy

Parte Radial

En este caso podemos escribir la integral igual que antes y por lo tanto nos da la misma respuesta que es

$$3a_0$$

Parte Angular

En este nuevo caso el cambio es simplemente *cos* por *sin* en el script anterior como:

```
import sympy as sp

theta, phi = sp.symbols('theta phi', real=True)

norm_const = sp.sqrt(3/(8*sp.pi))

Y = - norm_const * sp.exp(-sp.I*phi) * sp.sin(theta)
Y_abs2 = Y * sp.conjugate(Y)
Y_abs2 = sp.simplify(Y_abs2)
expr = Y_abs2 * sp.sin(theta)**2 * sp.sin(phi)

integral = sp.integrate(expr, (phi, 0, 2*sp.pi), (theta, 0, sp.pi))

sp.print_latex(integral)
```

lo que nos da como resultado

$$0$$

Por lo tanto el resultado es el mismo que en anterior como:

$$\begin{aligned}\langle x\rangle &= \int_0^\infty N^2 r^3 e^{-\frac{r}{a_0}} dr \int_0^\pi \int_0^{2\pi} \left|Y_1^{-1}\right|^2 \sin^2\theta \sin\phi d\phi d\theta \\ &= (3a_0)(0) \\ &= 0\end{aligned}$$

Chapter 4

Este problema podemos simplemente aplicar el operador como

$$L_+ = \hbar e^{i\phi}\left(\frac{\partial}{\partial\theta} + i\cot\theta\frac{\partial}{\partial\phi}\right)$$
$$Y_2^1 = -\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{i\phi}$$
$$L_+Y_2^1 = \hbar e^{i\phi}\left(\frac{\partial Y_2^1}{\partial\theta} + i\cot\theta\frac{\partial Y_2^1}{\partial\phi}\right)$$

Ahora, miremos cada una de estas derivadas parciales por aparte

$$\begin{aligned}\frac{\partial Y_2^1}{\partial\theta} &= \frac{\partial}{\partial\theta}\left(-\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{i\phi}\right)\\ &= -\sqrt{\frac{15}{8\pi}}e^{i\phi}\frac{\partial}{\partial\theta}\left(\sin\theta\cos\theta\right)\\ &= -\sqrt{\frac{15}{8\pi}}e^{i\phi}\left(\frac{\partial\sin\theta}{\partial\theta}\cos\theta + \sin\theta\frac{\partial\cos\theta}{\partial\theta}\right)\\ &= -\sqrt{\frac{15}{8\pi}}e^{i\phi}\left(\cos^2\theta - \sin^2\theta\right)\\ &= \sqrt{\frac{15}{8\pi}}e^{i\phi}\left(\sin^2\theta - \cos^2\theta\right)\end{aligned}$$

Y para el otro caso

$$\begin{aligned}\frac{\partial Y_2^1}{\partial\phi} &= \frac{\partial}{\partial\phi}\left(-\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta e^{i\phi}\right)\\ &= -\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta\left(\frac{\partial e^{i\phi}}{\partial\phi}\right)\\ &= -\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta ie^{i\phi}\end{aligned}$$

$$i\cot\theta\frac{\partial Y_2^1}{\partial\phi} = i\frac{\cos\theta}{\sin\theta}\left(-\sqrt{\frac{15}{8\pi}}\sin\theta\cos\theta ie^{i\phi}\right)$$
$$= \sqrt{\frac{15}{8\pi}}\cos^2\theta e^{i\phi}$$

Con esto podemos retornar a nuestra expresión original con lo que tenemos

$$L_+Y_2^1 = \hbar e^{i\phi}\left(\sqrt{\frac{15}{8\pi}}e^{i\phi}\left(\sin^2\theta - \cos^2\theta\right) + \sqrt{\frac{15}{8\pi}}\cos^2\theta e^{i\phi}\right)$$
$$= \hbar e^{i\phi}\left(\sqrt{\frac{15}{8\pi}}e^{i\phi}\left(sin^2\theta - \cos^2\theta + \cos^2\theta\right)\right)$$
$$= \sqrt{\frac{15}{8\pi}}\hbar e^{2i\phi}sin^2\theta$$

ahora, tomando en cuenta que

$$Y_2^2 = \sqrt{\frac{15}{32\pi}}\sin^2\theta e^{2i\phi}$$

Ahora bien, note que:

$$\sqrt{\frac{15}{18\pi}} = \sqrt{\frac{15}{18\pi}\frac{4}{4}}$$
$$= 2\sqrt{\frac{15}{32\pi}}$$

Lo que entonces nos deja con la expresión como:

$$L_+Y_2^1 = \sqrt{\frac{15}{8\pi}}\hbar e^{2i\phi}sin^2\theta$$
$$= 2\sqrt{\frac{15}{32\pi}}\hbar e^{2i\phi}sin^2\theta$$
$$= 2\hbar\sqrt{\frac{15}{32\pi}}e^{2i\phi}sin^2\theta$$

Que note que esto es Y_2^2 con lo que este termino seria:

$$L_+Y_2^1 = 2\hbar Y_2^2$$

Chapter 5

Notas

Durante el desarrollo de este trabajo se hizo uso de multiples scripts de sympy. Intentamos que cada uno de estos fuera autoconclusivo de modo que no se tenia que ver ningun otro para poder llegar al resultado esperado. Llegando incluso al punto de copiar las mismas funciones de un archivo a otro. Tambien creemos que se hicieron lo suficientemente facil como para que con solo mirarlos se pueda entender lo que hace. Sin embargo, entendemos que puede resultar interesante revisar 1 a 1 cada uno de esotos scripts para confirmar sus resultados por lo que si lo desea puede verificar cada uno de ellos en el siguiente repositorio: https://github.com/S1e7J/Cuantica_Tarea_3. Muchas gracias por su atención