

HW #6

2.1 Spin the N₂

1. Calculate the moment of inertia of ¹⁴N-¹⁶O, in amu Å², the rotational energy constant, $B = \frac{\hbar^2}{2I}$, in kJ mol⁻¹, and the rotational spectral constant, $\tilde{B} = B/hc$ in cm⁻¹.

Sol: $I = \mu r^2$ moment of inertia

$$^{14}\text{N} = 14.006 \text{ amu}$$

$$^{16}\text{O} = 15.999 \text{ amu}$$

$$\mu = \frac{^{14}\text{N} \cdot ^{16}\text{O}}{^{14}\text{N} + ^{16}\text{O}} \text{ reduced mass}$$

$$\mu = \frac{(14.006 \text{ amu})(15.999 \text{ amu})}{(14.006 \text{ amu}) + (15.999 \text{ amu})}$$

$$\mu = 7.468 \text{ amu}; I = (7.468 \text{ amu})(1.15077 \text{ Å})^2 \\ = 9.890 \text{ amu Å}^2$$

Next, $B = \frac{\hbar^2}{2I}$ rotational energy constant

$$B = \frac{(1.05457 \times 10^{-34} \text{ J.s})^2}{2(9.890 \text{ amu Å}^2)}$$

$$1 \text{ amu} = 1.66054 \times 10^{-27} \text{ kg}$$

$$1 \text{ \AA} = 10^{-10} \text{ m}$$

$$B = (1.05457 \times 10^{-34} \text{ J.s})^2$$

$$\underline{2(9.890 \cdot 1.66054 \times 10^{-27} \text{ kg} \times 10^{-20} \text{ m})}$$

$$B \approx 3.3859 \times 10^{-23} \text{ J}$$

$$3.3859 \times 10^{-23} \text{ J} \times 6.022 \times 10^{23} \text{ mol}^{-1} \approx 20.390 \text{ J/mol}$$

$$\underline{20.390 \text{ J/mol}} \approx 0.0204 \text{ kJ/mol}$$

Next,

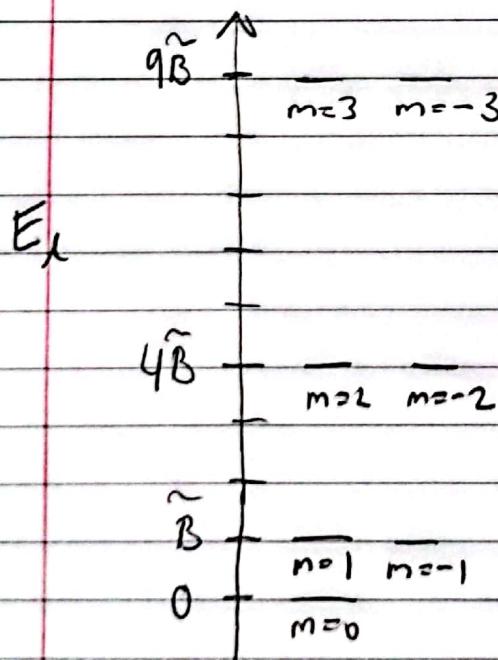
$$\underline{\underline{B}} = \frac{B}{hc} = \frac{3.3859 \times 10^{-23} \text{ J}}{(6.626 \times 10^{-34} \text{ J.s})(3 \times 10^8 \text{ cm/s})}$$
$$= 1.703 \text{ cm}^{-1}$$

$$0.0204 \text{ kJ/mol} = 20.390 \times 10^{-3} \text{ J/mol}$$
$$(6.626 \times 10^{-34} \text{ J.s})(3 \times 10^8 \text{ cm/s})$$

2. Imagine that the NO molecule is adsorbed flat on a surface upon which it is free to rotate. Plot out the energies of the four lowest-energy rotational quantum states, in units of \tilde{B} , being sure to include appropriate quantum numbers and degeneracies. Also indicate the total rotational angular moment of each state, in units of \hbar .

$$E_J = \tilde{B} m^2 ; \quad l_z = m\hbar$$

m	E_J	degeneracy	l_z
0	0	1	0
1	\tilde{B}	2	\hbar
2	$4\tilde{B}$	2	$2\hbar$
3	$9\tilde{B}$	2	$3\hbar$



3. Whether light can induce an NO to jump from some rotational level m_L to some other one m'_L is determined by whether the transition dipole moment integral $\langle \Psi_{m_L} | x | \Psi_{m'_L} \rangle$ is zero or non-zero. Find the

Selection rule on Δm_L that make the integral non-zero.

Recall that x can be written $r \cos \phi = r(e^{i\phi} + e^{-i\phi})/2$ in polar coordinates.

Sol:

If we regard NO as a 2-D rotor, we have the following wavefunction:

$$\Psi_{m_L}(\phi) = \frac{1}{\sqrt{2\pi}} e^{-im_L \phi} \cdot \langle \Psi_{m_L} | x | \Psi_{m'_L} \rangle$$

$$= \int_0^{2\pi} \frac{1}{\sqrt{2\pi}} e^{im_L \phi} \times \frac{1}{\sqrt{2\pi}} e^{-im'_L \phi} d\phi$$

Using polar coordinates

$$x = r \cos \phi$$

$$= \frac{r}{2\pi} \int_0^{2\pi} e^{im_L \phi - im'_L \phi} \cos \phi d\phi$$

$$\text{We have that: } \cos \phi = \frac{e^{-i\phi} + e^{i\phi}}{2}$$

$$\Rightarrow = \frac{r}{2\pi} \int_0^{2\pi} (e^{im_L \phi - im'_L \phi}) \left(\frac{e^{i\phi} + e^{-i\phi}}{2} \right) d\phi$$

Simplifying

$$= \frac{r}{4\pi} \int_0^{2\pi} \left(e^{i(\Delta m_l - 1)\phi} + e^{i(\Delta m_l + 1)\phi} \right) d\phi$$

$n \rightarrow$ integer ; when $n = 0$, $\int_0^{2\pi} e^{in\phi} d\phi \neq 0$

\therefore when $\Delta m_l = \pm 1$ the transition dipole moment integral is $\neq 0$.

4.

- Use your selection rule to determine the frequencies, in wavenumbers, of the four lowest-energy rotational transitions of an $^{14}\text{N} = ^{16}\text{O}$ adsorbed flat on a surface.

Table 2: Transitions

	ΔE	$\tilde{\nu} (\text{cm}^{-1})$
$m=0 \rightarrow 1$	\tilde{B}	1.703 cm^{-1}
$m=1 \rightarrow 2$	$3\tilde{B}$	5.109 cm^{-1}
$m=2 \rightarrow 3$	$5\tilde{B}$	8.515 cm^{-1}
$m=3 \rightarrow 4$	$7\tilde{B}$	11.921 cm^{-1}

5. Use your selection rule to determine the change in angular momentum of the " $N=0$ " in each allowed transition. Compare your result to the angular momentum of a photon, \hbar .

We have $\Delta m_l = \pm 1$

$$\Rightarrow l_m = m\hbar$$

$$l_{m+1} = \hbar(m+1)$$

$$\Rightarrow \Delta l = (m+1)\hbar - m\hbar$$

$$\therefore \Delta l = \pm \hbar$$

Result is equivalent to angular momentum of a photon.

1 Chem 30324, Spring 2024, Homework 6

1.1 Quantum mechanics of an H atom:

- 1.1.1 Consider an excited hydrogen atom with a $2s^1$ electron configuration. The $2s$ radial wavefunction is given by

$$R_{2,0} = \frac{1}{\sqrt{2a_0^3}}(1 - \rho/2)e^{-\rho/2}, \rho = r/a_0$$

where a_0 = bohr radius. (*Hint:* It's easiest to solve the following using a_0 as the unit of length.)

- 1.1.2 6. Provide a complete set of possible quantum numbers for an electron with the $2s$ wavefunction.

Since this is the $2s$ orbital, $n = 2$. S orbitals have $l = 0$, which makes $m_l = 0$ as well. Spin can be either positive or negative, so $m_s = \pm 1/2$

- 1.1.3 7. Provide a hand sketch of the 3-D shape of a $2s$ wavefunction (“orbital”). Be sure to indicate the *sign* of the wavefunction in each region and the location(s) of nodes. How many radial and how many angular nodes does a $2s$ wavefunction have?

Solutions have $n - 1$ radial nodes and l angular nodes, so this wavefunction has one radial node and no angular nodes. By observing the probability distribution function (see below), we can see that the radial node is at $r = 2a_0$

- 1.1.4 8. Plot the radial probability function $P_{20}(r) = r^2|R_{2,0}(r)|^2$. Be sure to label both axes appropriately. (*Hint:* Stick with a_0 as the length unit.)

```
[35]: import numpy as np
import matplotlib.pyplot as plt

#Setting a0 to 1 to imply lengths are in units of a0
rho = np.linspace(0,15,1000)

def R(rho):
    return ((1/np.sqrt(2))*(1-rho/2)*np.exp(-rho/2))

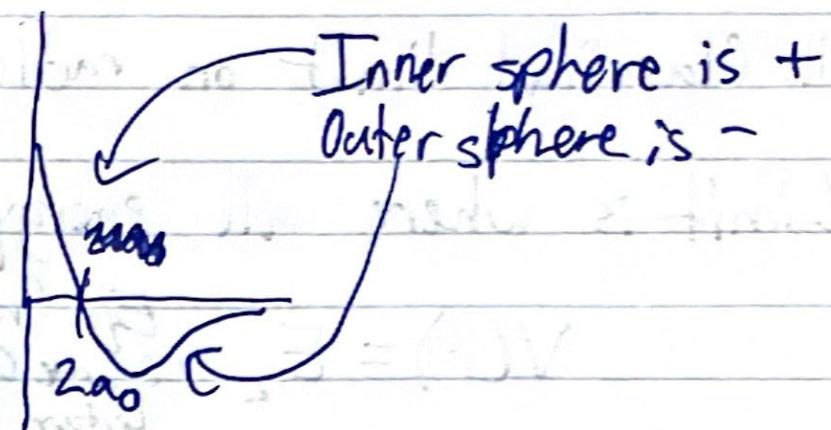
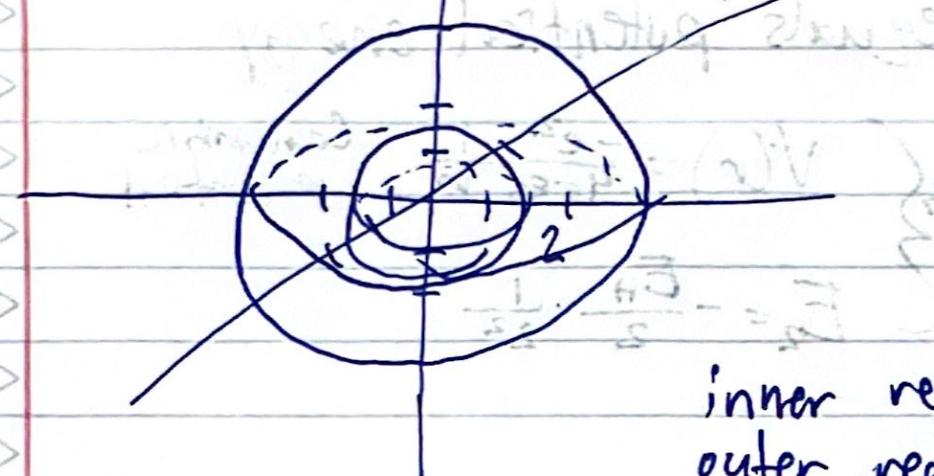
Prob_20 = rho**2*(R(rho))**2

plt.plot(rho,Prob_20)
plt.xlabel('r/a0')
plt.ylabel('r**2*R(r)**2 (1/a0)')
plt.axvline(6,0,4e9, color ='b')# <r>, solved below

plt.axvline(np.sqrt(5)+3,0,4e9,color='r')# most probable r, solved below

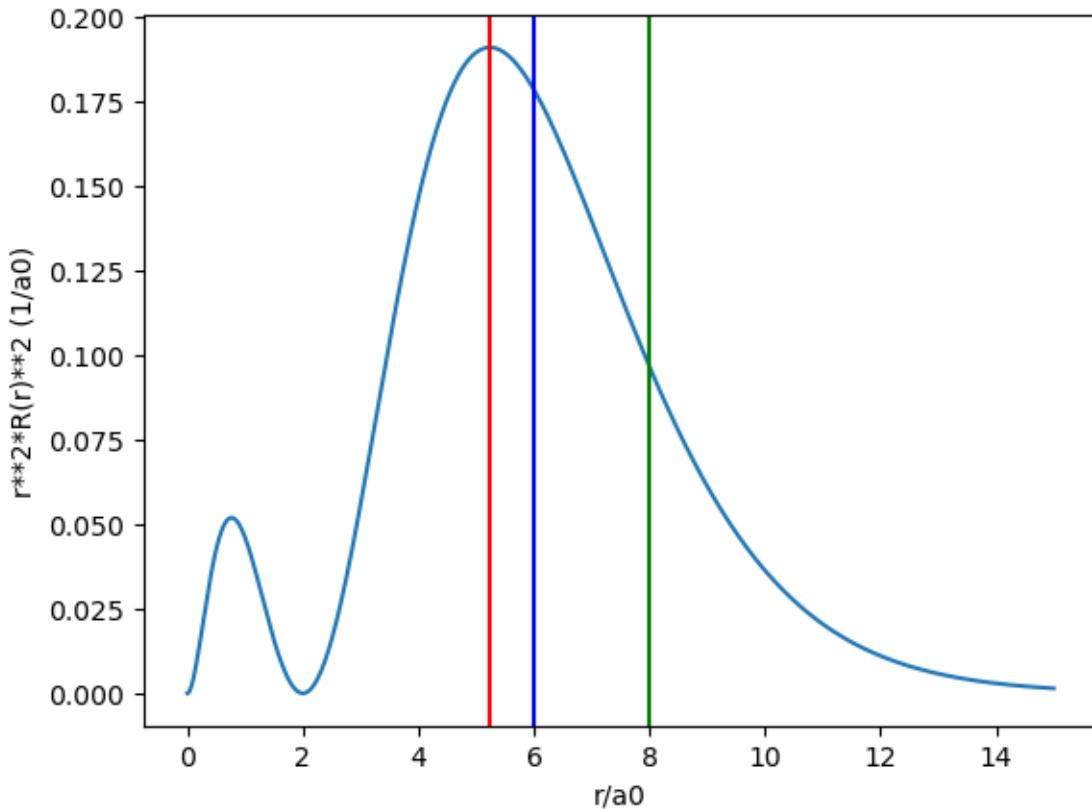
plt.axvline(8,0,4e9,color='g')# Classical limit of r, solved below
```

► 7. Hand sketch of 2s "orbital" shape



inner region is positive,
outer region is negative

[35]: <matplotlib.lines.Line2D at 0x1ece1973790>



- 1.1.5 9. Determine and indicate on your plot $\langle r \rangle$, the expectation value of the distance of the electron from the nucleus. (Hint: the course outline gives an expression for this expectation value.)

From course outline notes, $\langle r \rangle = (\frac{3}{2}n^2 - l(l + 1))a_0$, since $l = 0$ and $n = 2$, this simplifies to $= (\frac{3}{2} * 4 - 0)a_0 = 6a_0$, shown as a blue line on the graph.

- 1.1.6 10. Determine and indicate on your plot r_{MP} , the most probable distance of the electron from the nucleus.

Using Wolfram-Alpha, derivative of the probability function is $-\frac{1}{8}e^{-r} * r(-8 + 16r - 8r^2 + r^3)$, Wolfram provides the roots for this function but we can confirm which one we want, which should be around 5.

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$$\frac{d}{dr} \left(r^2 * \left(\frac{1}{\sqrt{2}} \left(1 - \frac{r}{2} \right) * e^{-\frac{r}{2}} \right)^2 \right)$$

✖ =

NATURAL LANGUAGE

MATH INPUT

★ √ ⌂ (:) √ω ⌂ω ...

Assuming multiplication | Use a list instead

Derivative

Step-by-step solution

$$\frac{d}{dr} \left(r^2 \left(\frac{(1 - \frac{r}{2}) \exp(-\frac{r}{2})}{\sqrt{2}} \right)^2 \right) = -\frac{1}{8} e^{-r} r (r^3 - 8r^2 + 16r - 8)$$

Roots

$$r = 0$$

$$r = 2$$

$$r = 3 - \sqrt{5}$$

$$r = 3 + \sqrt{5}$$

```
[21]: from scipy.optimize import fsolve  
  
deriv = lambda r: -1/8*np.exp(-r)*r*(-8+16*r-8*r**2+r**3) # Defining derivative  
  
root = fsolve(deriv, [5]) #will find route closest to the initial guess of 5  
print(root)
```

[5.23606798]

```
[22]: rt = np.sqrt(5)+3 # proving that this value is indeed the route  
print(rt)
```

5.23606797749979

Point is shown on the graph by a red line.

1.1.7 11. Determine and indicate on your plot the maximum classical distance of the electron from the nucleus in this orbital.

```
[29]: c = 14.39964# eV*A, potential energy constants bundled together
E_H = 27.212# eV
A_0 = 0.529177# A
r_cl = (8*c)/E_H
print(r_cl, 'Angstrom')

R_cl = r_cl/A_0
print(R_cl, '*a_0')
```

```
4.233320593855652 Angstrom
7.99981970844472 *a_0
```

See below for how this is derived, plotted as a green line on the graph

1.1.8 12. (Extra Credit) What is the probability of finding the electron beyond the classical distance? (Evaluate the necessary integral numerically.)

```
[31]: from scipy.integrate import quad

a= 8
b = np.inf
def Prob(R):
    return (1/np.sqrt(2)*(1-(R/(2)))*np.exp(-R/(2)))**2*R**2
prob,err = quad(Prob,a,b)
print(prob)
```

```
0.1855108332300911
```

Probability of being beyond the classical limit is about 18.55%

1.1.9 13. Can a 2s electron undergo an allowed transition to a lower energy electronic state? If so, what frequency (in cm^{-1}) of light would be emitted?

The only lower energy state would be the 1s orbital, however since quantum number l would remain at 0 this transition would violate the selection rule of $\Delta l = \pm 1$ and is therefore forbidden.

II. Classical limit on radius

Limit is where all energy equals potential energy

$$V(r) = E_2 \quad \left. \begin{array}{l} \text{Eqs. from} \\ \text{course outline} \\ \text{lecture II} \end{array} \right\} \quad V(r) = \frac{-e^2}{4\pi\epsilon_0 r}, \quad \begin{array}{l} \text{Coulombic} \\ \text{potential} \end{array}$$
$$\frac{-e^2}{4\pi\epsilon_0} \cdot \frac{1}{r} = -\frac{E_H}{2} \cdot \frac{1}{r}$$

*start at origin zero
end at origin zero*

$$\frac{-e^2}{4\pi\epsilon_0} = -\frac{E_H r}{8}$$

$$\frac{8e^2}{4\pi\epsilon_0 E_H} = r$$

$$8 \cdot \left(\frac{e^2}{4\pi\epsilon_0} \right) \frac{1}{E_H} =$$

Values provided in Table I
of course outline

14. Suppose in a fit of panic you forget the 1s radial function when asked on an exam. Not wanting to leave the answer blank, you decide to guess something, and liking bell-shaped curves, you guess $R_1(r) = e^{-r}$. Normalize this guess. Do not forget to include the r^2 Jacobian integration factor.

$$\int_0^\infty r^2 e^{-r^2} dr = N^2$$

Using wolfram to solve integral

$$N^2 = \frac{\sqrt{\pi}}{4}$$

$$N = \left(\frac{\sqrt{\pi}}{4}\right)^{1/2} = \frac{\sqrt{\pi}}{2}$$

$$\Rightarrow \tilde{R}(r) = \frac{2}{\sqrt[4]{\pi}} e^{-r^2}$$

$$\left\{ -\frac{1}{2} \frac{d^2}{dr^2} - \frac{1}{r} \frac{d}{dr} - \frac{1}{r} + \frac{\lambda(\lambda+1)}{2r^2} \right\} R(r) = ER(r)$$

15. Calculate the expectation value of the energy of your normalized guess. Is it greater or less than the true value?

$$\langle \psi | \hat{H} | \psi \rangle$$

$$\hat{H} = -\frac{1}{2} \frac{d^2}{dr^2} - \frac{1}{r} \frac{d}{dr} - \frac{1}{r} + \frac{\lambda(\lambda+1)}{2r}$$

⇒ Let's calculate

$$\frac{d}{dr} e^{-r^2} = -2re^{-r^2}$$

$$\begin{aligned} \frac{d^2}{dr^2} e^{-r^2} &= -2(e^{-r^2} + r(-2re^{-r^2})) \\ &= (-2 + 4r^2)e^{-r^2} \end{aligned}$$

$$\hat{H} \cdot R = -\frac{1}{2}((-2 + 4r^2)e^{-r^2}) - \frac{1}{r}(-2re^{-r^2}) - \frac{1}{r}$$

Simplifying

$$= \left(3 - 2r^2 - \frac{1}{r}\right) e^{-r^2}$$

$$\Rightarrow \frac{2}{\sqrt[4]{\pi}} \int e^{-2r^2} (r^2) \left(3 - 2r^2 - \frac{1}{r}\right) dr$$

$$\Rightarrow \left(\frac{2}{\sqrt[4]{\pi}}\right)^2 \int_0^\infty e^{-2r^2} (3r^2 - 2r^4 - r) dr$$

$$\langle E \rangle = -0.095 E_H$$

Q6. What does the variation principle say about the expectation value of the energy of your guess as you vary a parameter γ in your guess, $R_{1,0} = e^{-\gamma r^2}$?

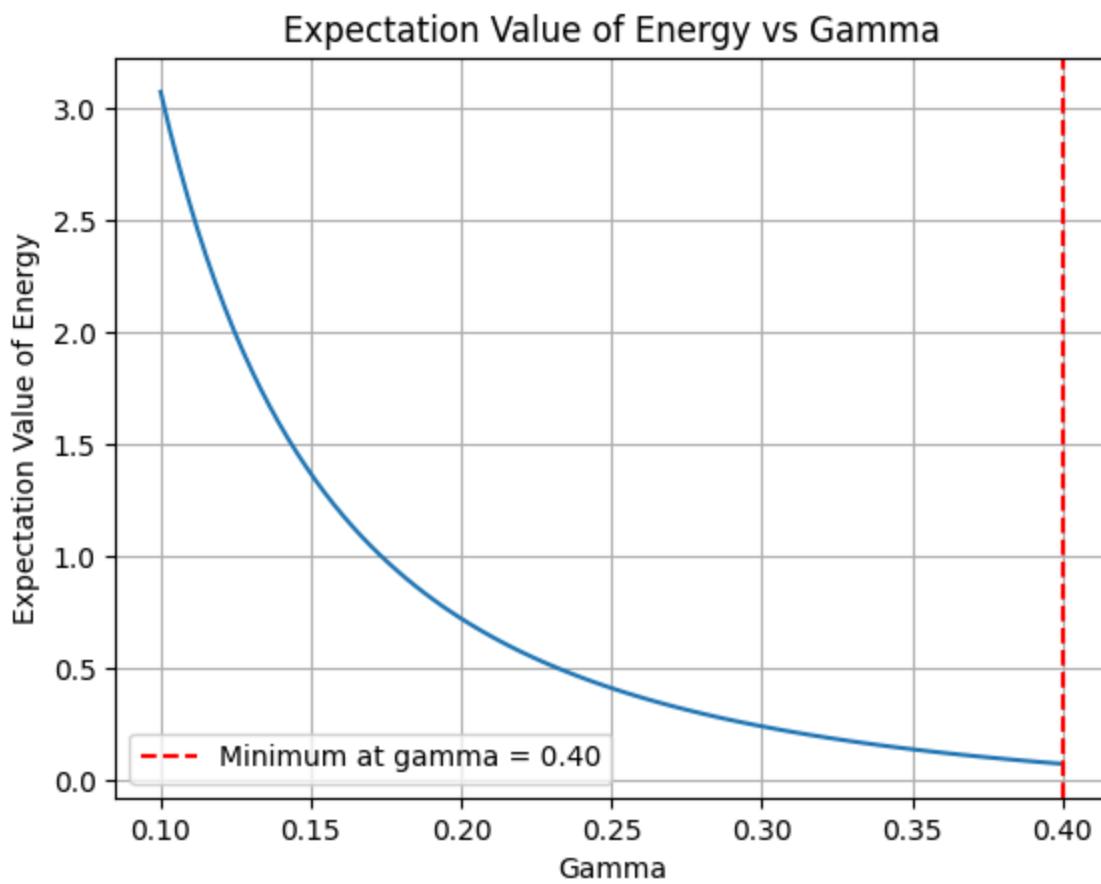
Variation principle states that the ground-state energy E_0 is always less than or equal to the expectation value of H calculated with:

$$E_0 \leq \langle \psi | \hat{H} | \psi \rangle$$

γ could have different values and execute a plot that could be $\langle \psi | \hat{H} | \psi \rangle$ versus γ and analyze the behavior.

16.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Define the potential energy function for the hydrogen atom
5 def potential_energy(r):
6     return -1 / r
7
8 # Define the trial wavefunction squared
9 def trial_wavefunction_squared(r, gamma):
10    return np.exp(-2 * gamma * r**2)
11
12 # Function to compute the expectation value of energy using numerical integration
13 def expectation_energy(gamma, num_points=1000):
14     # Define the range of integration (let's use a large range for better accuracy)
15     r_values = np.linspace(0.001, 100, num_points)
16     dr = r_values[1] - r_values[0] # Step size
17
18     # Compute the integrand for the kinetic energy
19     kinetic_energy_integrand = 0.5 * trial_wavefunction_squared(r_values, gamma)
20                 (-2 * gamma * r_values**2 + 4 * gamma**2 * r_valu
21
22     # Compute the integrand for the potential energy
23     potential_energy_integrand = trial_wavefunction_squared(r_values, gamma) * po
24
25     # Perform numerical integration using the trapezoidal rule
26     kinetic_energy_integral = np.trapz(kinetic_energy_integrand * r_values**2, dx
27     potential_energy_integral = np.trapz(potential_energy_integrand * r_values**2
28
29     # Total energy is the sum of kinetic and potential energy
30     total_energy = kinetic_energy_integral + potential_energy_integral
31
32     return total_energy
33
34 # Define the range of gamma values focusing around 0.25
35 gamma_values = np.linspace(0.1, 0.4, 100)
36
37 # Calculate expectation values of energy for each gamma
38 energy_values = [expectation_energy(gamma) for gamma in gamma_values]
39
40 # Find the index of the minimum energy value
41 min_energy_index = np.argmin(energy_values)
42 min_energy_gamma = gamma_values[min_energy_index]
43 min_energy = energy_values[min_energy_index]
44
45 # Plot <E> vs gamma
46 plt.plot(gamma_values, energy_values)
47 plt.xlabel('Gamma')
48 plt.ylabel('Expectation Value of Energy')
49 plt.title('Expectation Value of Energy vs Gamma')
50 plt.axvline(x=min_energy_gamma, color='r', linestyle='--', label=f'Minimum at gamma={min_energy_gamma:.3f}')
51 plt.legend()
52 plt.grid(True)
53 plt.show()
54
```



17. Determine the best value of gamma.

```
1 # Find the index of the minimum energy value
2 min_energy_index = np.argmin(energy_values)
3 min_energy_gamma = gamma_values[min_energy_index]
4 min_energy = energy_values[min_energy_index]
5
6 # Print the minimum energy and corresponding gamma value
7 print("Minimum energy:", min_energy)
8 print("Corresponding gamma:", min_energy_gamma)
9
```

→ Minimum energy: 0.07251477957033736
Corresponding gamma: 0.4