```python
import numpy as np
from scipy.special import sph_harm, genlaguerre
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Constants and quantum numbers
n = 2       # Principal quantum number
l = 1       # Azimuthal quantum number
m = 0       # Magnetic quantum number
a0 = 1.0    # Bohr radius in atomic units (set to 1 for simplicity)

# Define radial part of hydrogen wavefunction (in atomic units)
def R_nl(r, n, l):
    rho = 2 * r / (n * a0)
    norm = np.sqrt((2 / (n * a0))**3 * np.math.factorial(n - l - 1) / (2 * n *
np.math.factorial(n + l)))
    laguerre_poly = genlaguerre(n - l - 1, 2 * l + 1)
    return norm * np.exp(-rho / 2) * rho**l * laguerre_poly(rho)

# Define angular part using spherical harmonics
def Y_lm(theta, phi, l, m):
    return sph_harm(m, l, phi, theta)

# Define full wavefunction Ψ_nlm(r, θ, φ)
def psi_squared(r, theta, phi, n, l, m):
    R = R_nl(r, n, l)
    Y = Y_lm(theta, phi, l, m)
    return np.abs(R * Y)**2

# Create spherical coordinate grid
grid_size = 100
r = np.linspace(0, 20, grid_size)
theta = np.linspace(0, np.pi, grid_size)
phi = np.linspace(0, 2 * np.pi, grid_size)
r, theta, phi = np.meshgrid(r, theta, phi)

# Convert to Cartesian coordinates for 3D plotting
x = r * np.sin(theta) * np.cos(phi)
y = r * np.sin(theta) * np.sin(phi)
z = r * np.cos(theta)

# Compute probability density
prob_density = psi_squared(r, theta, phi, n, l, m)

# Normalize and threshold for visualization
threshold = np.max(prob_density) * 0.01
mask = prob_density > threshold

# Plotting
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.set_title(f'3D Hydrogen Atom Orbital | n={n}, l={l}, m={m}', fontsize=14)

# Plot isosurface using scatter
ax.scatter(x[mask], y[mask], z[mask], c=prob_density[mask], cmap='viridis', s=1, alpha=0.3)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_xlim([-10, 10])
ax.set_ylim([-10, 10])
ax.set_zlim([-10, 10])
plt.tight_layout()
plt.show()
```