

```
clc;
clear;

% Define permittivity of free space
epsilon_0 = 1e-9/(36*pi);

% Given charge density
charge_density = 2e-6;

% Bounds for spherical coordinates (rho, theta, phi)
rho_max = 3;
rho_min = 2;
theta_max = pi;
theta_min = 0;
phi_max = 2*pi;
phi_min = 0;

% Number of steps in each coordinate direction
num_rho_steps = 10;
num_theta_steps = 10;
num_phi_steps = 10;

% Compute step sizes in each direction
delta_rho = (rho_max - rho_min) / num_rho_steps;
delta_theta = (theta_max - theta_min) / num_theta_steps;
delta_phi = (phi_max - phi_min) / num_phi_steps;

% Initialize total stored energy variable
total_energy = 0;

% Triple nested loop to iterate over rho, theta, and phi
for theta_index = 1:num_theta_steps
    for phi_index = 1:num_phi_steps
        for rho_index = 1:num_rho_steps
            % Compute midpoints for integration
            rho = rho_min + delta_rho * 0.5 + (rho_index - 1) * delta_rho;
            theta = theta_min + delta_theta * 0.5 + (phi_index - 1) * delta_theta;
            phi = phi_min + delta_phi * 0.5 + (theta_index - 1) * delta_phi;

            % Compute electric field magnitude
            E_field = (charge_density / (epsilon_0 * rho^2));

            % Compute volume element in spherical coordinates
            dV = rho^2 * sin(theta) * delta_theta * delta_phi * delta_rho;

            % Compute energy stored in the current volume element
            dW = 0.5 * epsilon_0 * E_field^2 * dV;

            % Accumulate total energy
            total_energy = total_energy + dW;
        end
    end
end

% Display final stored energy value
disp(total_energy)
```