In [2]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:
```python
# Task 1
```

Define the probability density function (PDF) for the nearest star distance.

For a uniform Poisson process in 3D, the probability that no star is found within a sphere of radius R is:

$$P(\text{no stars in sphere}) = \exp\left(-\frac{4}{3}\pi n R^3\right)$$

Therefore, the cumulative probability that the nearest star is within distance R is:

$$\text{CDF}(R) = 1 - \exp\left(-\frac{4}{3}\pi n R^3\right)$$

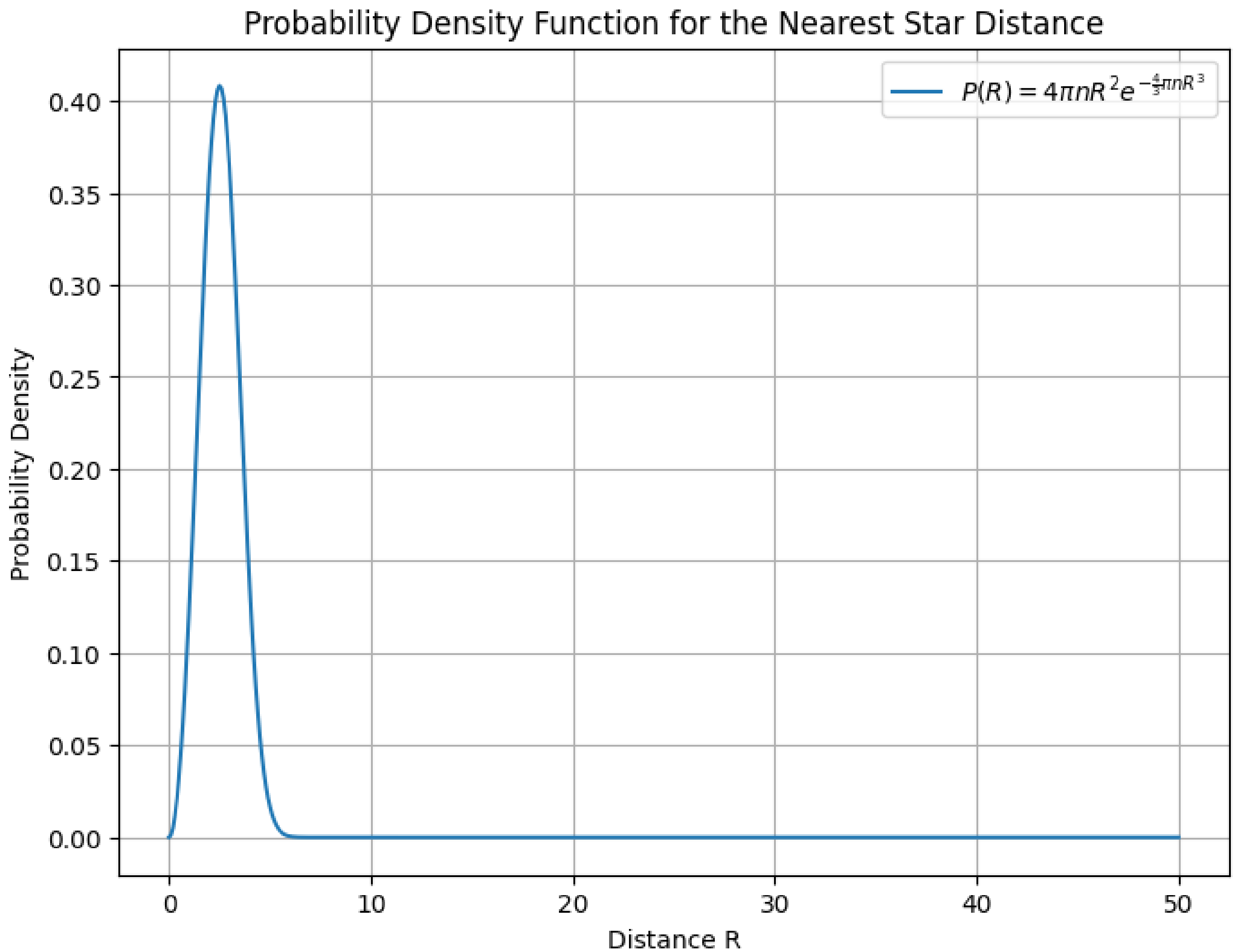
Differentiating the CDF with respect to R gives the PDF:

$$\text{PDF}(R) = \frac{d}{dR}\left[1 - \exp\left(-\frac{4}{3}\pi n R^3\right)\right] = 4\pi n R^2 \exp\left(-\frac{4}{3}\pi n R^3\right)$$

In [5]:
```python
n = 0.01  # You can adjust this value as needed


def nearest_star_pdf(R, n):
    return 4 * np.pi * n * R**2 * np.exp(- (4/3) * np.pi * n * R**3)

R_values = np.linspace(0, 50, 500)  # Adjust the range as needed
pdf_values = nearest_star_pdf(R_values, n)
plt.figure(figsize=(8, 6))
plt.plot(R_values, pdf_values, label=r'$P(R) = 4 \pi n R^2 e^{-\frac{4}{3}\pi n R^3}$')
plt.xlabel('Distance R')
plt.ylabel('Probability Density')
plt.title('Probability Density Function for the Nearest Star Distance')
plt.legend()
plt.grid(True)
plt.show()
```



Task 2

Consider the driven, damped harmonic oscillator:

$$\frac{d^2x}{dt^2} + \gamma\frac{dx}{dt} + \omega_0^2 x = Fe^{i\omega_f t}$$

Taking the Fourier transform yields:

$$(-\omega_f^2 + i\gamma\omega_f + \omega_0^2)\tilde{x} = F$$

Thus, the Fourier amplitude is:

$$\tilde{x} = \frac{F}{\omega_0^2 - \omega_f^2 + i\gamma\omega_f}$$

Defining the susceptibility as

$$\chi(\omega_f) = \frac{1}{\omega_0^2 - \omega_f^2 + i\gamma\omega_f},$$

its imaginary part is:

$$\mathrm{Im}[\chi(\omega_f)] = \frac{\gamma\omega_f}{(\omega_0^2 - \omega_f^2)^2 + \gamma^2\omega_f^2}$$

Therefore, the energy absorbed per cycle is given by:

$$E = F\pi\ \mathrm{Im}[\chi(\omega_f)] = \frac{F\pi\gamma\omega_f}{(\omega_0^2 - \omega_f^2)^2 + \gamma^2\omega_f^2}$$

In [7]:
```
# Task 3
```

Question 1: Define the transition matrix P for the 3-spin system. Label the states as follows:

$$\begin{aligned} 1 &: \ |\uparrow\uparrow\uparrow\rangle, \\ 2 &: \ |\uparrow\uparrow\downarrow\rangle, \\ 3 &: \ |\uparrow\downarrow\uparrow\rangle, \\ 4 &: \ |\uparrow\downarrow\downarrow\rangle, \\ 5 &: \ |\downarrow\uparrow\uparrow\rangle, \\ 6 &: \ |\downarrow\uparrow\downarrow\rangle, \\ 7 &: \ |\downarrow\downarrow\uparrow\rangle, \\ 8 &: \ |\downarrow\downarrow\downarrow\rangle. \end{aligned}$$

Only pairs of neighboring spins that are anti-aligned (i.e. one up and one down) can "flip" via the hopping operators. For example, if sites i and $i+1$ form the pair $(\uparrow,\downarrow)$ they can flip to $(\downarrow,\uparrow)$ and vice-versa. With periodic boundary conditions the pairs are $(1,2)$, $(2,3)$ and $(3,1)$. Assign equal probability to each allowed move. Then the transition matrix is:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Question 2: The stationary distribution π satisfies

$$\pi P = \pi, \quad \sum_{i=1}^{8} \pi_i = 1.$$

Notice that states 1 and 8 have no allowed moves (they are "frozen"), while the remaining states form two connected blocks:

- Block A: states 2, 3, 5
- Block B: states 4, 6, 7

Within each block the transitions are symmetric, so the stationary probabilities in each block are uniform. In other words:

$$\pi_2 = \pi_3 = \pi_5, \qquad \pi_4 = \pi_6 = \pi_7.$$

Alternatively, if energy differences are incorporated (via Boltzmann factors at temperature T), then

$$\pi_i = \frac{e^{-E_i/(k_B T)}}{Z}, \qquad Z = \sum_{i=1}^{8} e^{-E_i/(k_B T)}.$$

Question 3: Power iteration is given by

$$\pi^{(k+1)} = \pi^{(k)} P.$$

For the following initial guesses:

If

$$\pi^{(0)} = (1, 0, 0, 0, 0, 0, 0, 0)$$

the chain remains in state 1.

If

$$\pi^{(0)} = \left(0, \frac{1}{2}, 0, 0, \frac{1}{2}, 0, 0, 0\right),$$

the iteration is confined to Block A and converges to

$$\pi_2 = \pi_3 = \pi_5 = \frac{1}{3} \quad \text{(within Block A)}.$$

If

$$\pi^{(0)} = \left(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right)$$

the iteration converges to a mixture of the stationary distributions in each disjoint block.

Question 4: Markov Chain in Magnon Basis

For a chain of $N = 3$ spins, the allowed magnon momenta are quantized as

$$p_k = \frac{2\pi k}{3}, \quad k = 0, 1, 2,$$

with corresponding energies

$$E_k = 2J \sin^2\left(\frac{\pi k}{3}\right).$$

Thus,

$$E_0 = 0, \quad E_1 = 2J \sin^2\left(\frac{\pi}{3}\right) = \frac{3J}{2}, \quad E_2 = \frac{3J}{2}.$$

Assuming Boltzmann-type transitions between magnon states, we take the transition probability from state i to state j as

$$P_{ij} \sim e^{-\frac{E_j - E_i}{k_B T}}.$$

Normalizing each row, we define:

For state 0:

$$P_{0j} = \frac{e^{-\frac{E_j - 0}{k_B T}}}{Z_0}, \quad Z_0 = e^0 + 2e^{-\frac{3J}{2k_B T}} = 1 + 2e^{-\frac{3J}{2k_B T}}.$$

For state 1 (and similarly state 2):

$$P_{1j} = \frac{e^{-\frac{E_j - E_1}{k_B T}}}{Z_1}, \quad Z_1 = e^{\frac{3J}{2k_B T}} + 1 + 1 = e^{\frac{3J}{2k_B T}} + 2.$$

Thus, the transition matrix in the magnon basis is:

$$P = \begin{pmatrix} \frac{1}{1+2e^{-\frac{3J}{2k_B T}}} & \frac{e^{-\frac{3J}{2k_B T}}}{1+2e^{-\frac{3J}{2k_B T}}} & \frac{e^{-\frac{3J}{2k_B T}}}{1+2e^{-\frac{3J}{2k_B T}}} \\ \frac{e^{\frac{3J}{2k_B T}}}{e^{\frac{3J}{2k_B T}}+2} & \frac{1}{e^{\frac{3J}{2k_B T}}+2} & \frac{1}{e^{\frac{3J}{2k_B T}}+2} \\ \frac{e^{\frac{3J}{2k_B T}}}{e^{\frac{3J}{2k_B T}}+2} & \frac{1}{e^{\frac{3J}{2k_B T}}+2} & \frac{1}{e^{\frac{3J}{2k_B T}}+2} \end{pmatrix}.$$

The key difference relative to the site-basis transition matrix is that here the probabilities are weighted by energy differences (via Boltzmann factors), reflecting the thermal population of the magnon states.

Question 5: Stationary Distribution in the Magnon Basis

The stationary distribution π satisfies

$$\pi P = \pi, \quad \sum_{k=0}^{2} \pi_k = 1.$$

Because the system obeys detailed balance, the stationary distribution is given by the Boltzmann distribution:

$$\pi_k = \frac{e^{-\frac{E_k}{k_B T}}}{\sum_{k'=0}^{2} e^{-\frac{E_{k'}}{k_B T}}}.$$

For $N = 3$ with $E_0 = 0$ and $E_1 = E_2 = \frac{3J}{2}$, we have:

$$\pi_0 = \frac{1}{1 + 2e^{-\frac{3J}{2k_B T}}}, \quad \pi_1 = \pi_2 = \frac{e^{-\frac{3J}{2k_B T}}}{1 + 2e^{-\frac{3J}{2k_B T}}}.$$

Question 6: Power Iteration in the Magnon Basis

Power iteration evolves the distribution by

$$\pi^{(k+1)} = \pi^{(k)} P.$$

Consider three initial guesses:

Initial Guess 1: (\pi^{(0)} = (0,1,0)) (i.e. starting in state (|k=1\rangle)). Iteration will drive the distribution toward the stationary state:

$$\pi_0 = \frac{1}{1 + 2e^{-\frac{3J}{2k_BT}}}, \quad \pi_1 = \pi_2 = \frac{e^{-\frac{3J}{2k_BT}}}{1 + 2e^{-\frac{3J}{2k_BT}}}.$$

Initial Guess 2:

$$\pi^{(0)} = \left(0, \frac{1}{2}, \frac{1}{2}\right).$$

The iteration remains in the subspace of states (1) and (2) and converges to the same stationary distribution.

Initial Guess 3:

$$\pi^{(0)} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$$

.

Again, repeated multiplication by (P) leads to convergence to the Boltzmann stationary distribution.

Question 7: Master Equation Evolution

The continuous-time master equation is given by:

$$\frac{d\pi_i}{dt} = \sum_j \left(Q_{ji}\pi_j - Q_{ij}\pi_i\right).$$

To relate the discrete-time transition matrix P to the continuous-time transition rate matrix Q, we use:

$$P \approx e^{Q\Delta t} \quad \Longrightarrow \quad Q \approx \frac{1}{\Delta t}\ln(P).$$

Once Q is determined, the master equation in matrix form reads:

$$\frac{d\pi}{dt} = \pi Q,$$

with an initial condition such as $\pi(0) = (0, 1, 0)$ (corresponding to starting in $|k = 1\rangle$). This system can be solved numerically using standard ODE integrators (for example, with Python's `scipy.integrate.solve_ivp` *or* `odeint`*).*