```python
# Exercise 2.2 (c).
import matplotlib.animation as animation
import matplotlib.pyplot as plt
import numpy as np

# Initialize lattice
lattice_size = 200
lattice = np.sign(np.random.rand(lattice_size,lattice_size) - 0.5)
new_lattice = lattice.copy()
ten_percent = int(lattice_size*lattice_size/10)

# Constants
J = 1
T = 6
H = 1
beta = 1/T
iterations = 50000
magnetization_array = np.zeros((1,iterations))
energies = np.array([0.1, 0.2, 0.3, 0.4])

# Animation
fig1, ax = plt.subplots()
ims = []
im = ax.imshow(lattice.copy())
ims.append([im])

# Plotting
fig2, axs = plt.subplots(1, 4, figsize=(12,12))
time_0 = lattice.copy()

for time_step in range(iterations):

    # Update randomly 10% of the cells
    for update in range(ten_percent):

        i = np.random.randint(lattice_size)
        j = np.random.randint(lattice_size)

        M = 0

        # Due to boundaries
        if i > 0:
            M += lattice[i-1,j]
        if i < lattice_size-1:
            M += lattice[i+1,j]
        if j > 0:
            M += lattice[i,j-1]
        if j < lattice_size-1:
            M += lattice[i,j+1]

        E_plus = -H-J*M
        E_minus = H+J*M

        prob_plus = np.exp(-beta*E_plus) / (np.exp(-beta*E_plus) + np.exp(-beta*E_minus))
        rnd = np.random.rand()

        if rnd < prob_plus:
            new_lattice[i,j] = 1
```

```
        else:
            new_lattice[i,j] = -1

    lattice = new_lattice.copy()

     # Snapshots of certain time steps
    if time_step == 100-1:
        time_1 = lattice.copy()
    elif time_step == 10000-1:
        time_2 = lattice.copy()
    elif time_step == 50000-1:
        time_3 = lattice.copy()

    # Computing magnetization per unit volume to measure the state of the magnetic property
    m = 0
    for i in range(lattice_size):
        for j in range(lattice_size):
            m += (lattice[i,j] / np.power(lattice_size, 2))

    magnetization_array[0,time_step] = m

    # Images for animation
    # im = ax.imshow(lattice.copy(), animated=True)
    # ims.append([im])

    if time_step % 100 == 0:
        print(time_step)

# Plotting
axs[0].imshow(time_0)
axs[0].yaxis.set_ticks([])
axs[0].xaxis.set_ticks([])

axs[1].imshow(time_1)
axs[1].yaxis.set_ticks([])
axs[1].xaxis.set_ticks([])

axs[2].imshow(time_2)
axs[2].yaxis.set_ticks([])
axs[2].xaxis.set_ticks([])

axs[3].imshow(time_3)
axs[3].yaxis.set_ticks([])
axs[3].xaxis.set_ticks([])

axs[0].set_ylabel('T = 6')
axs[0].set_title('t = 0')
axs[1].set_title('t = 100')
axs[2].set_title('t = 10000')
axs[3].set_title('t = 50000')

plt.subplots_adjust(wspace=0.1, hspace=0.05)
plt.savefig('22c.png', bbox_inches='tight')

# ani = animation.ArtistAnimation(fig1, ims, interval=5, blit=True)
# writergif = animation.PillowWriter(fps=30)
# ani.save("22c.gif", writer=writergif)

plt.axis('off')
```

```
plt.show()
```