HW3

Ising Model with Visualization

陳凱騫

2024-12-17

Table of contents

模型邏輯與程式碼 1
2D Ising Model Simulation 1
1. Simulation of Lattice Evolution 1
2. Influence of Temperature 3
3. Different Initial Conditions 3
4. Different Boundary Conditions 4
5. Results 5

模型邏輯與程式碼

```r
# 載入所需套件
library(ggplot2)
library(reshape2)
library(gridExtra)
```

2D Ising Model Simulation

This document describes the implementation and visualization of a 2D Ising model simulation using the Metropolis algorithm. The simulation includes: - Evolution of the lattice over time - Influence of temperature (T) - Different initial conditions - Different boundary conditions

1. Simulation of Lattice Evolution

The 2D Ising model simulates spins (+1 or -1) on a lattice, where the state of each spin evolves based on its interaction with neighboring spins and the temperature T.

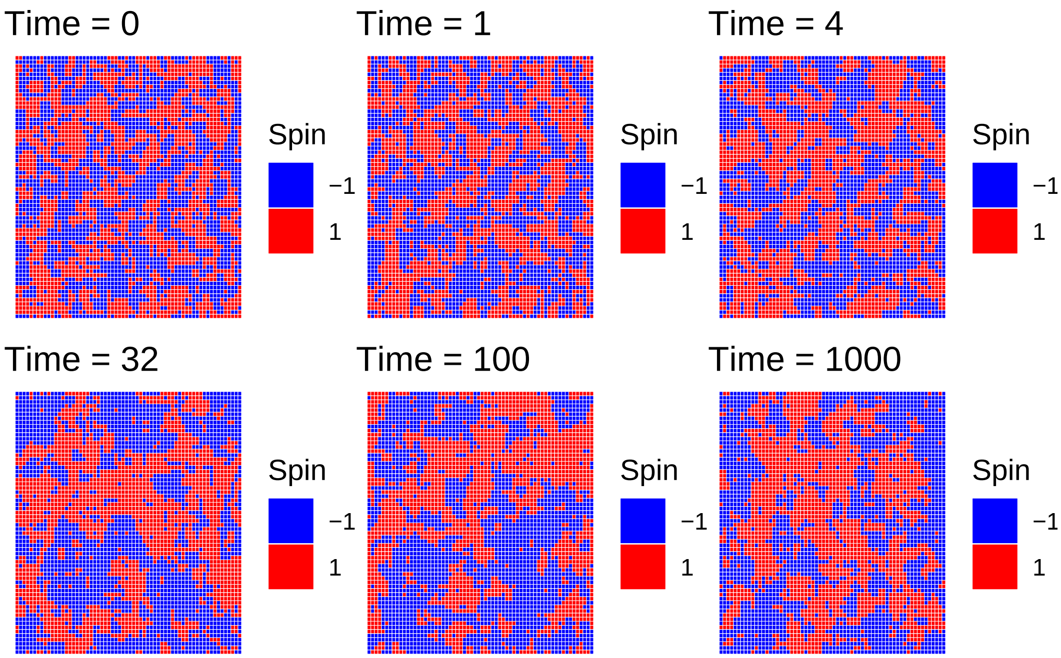
Key Steps:

1. Initialize the lattice with random spins.
2. Perform the Metropolis update rule for each spin.
3. Visualize the lattice at different time steps.

Code:

```r
initialize_lattice <- function(L) {
  matrix(sample(c(-1, 1), L * L, replace = TRUE), L, L)
}
```

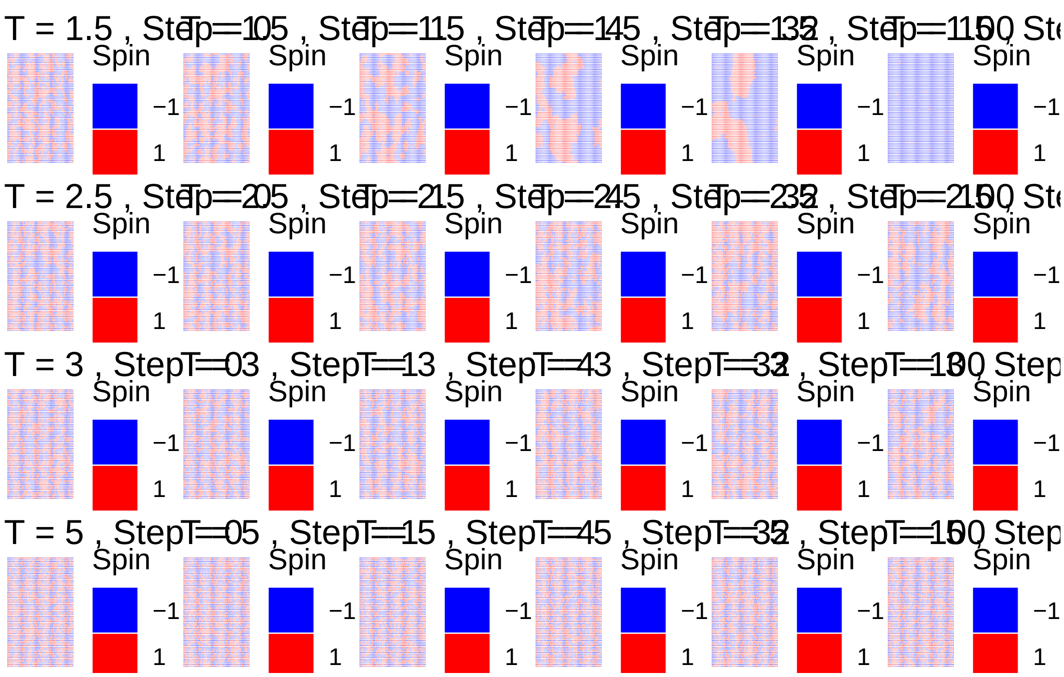
```r
metropolis_step <- function(lattice, beta, L) {
  for (k in 1:(L * L)) {
    i <- sample(1:L, 1)
    j <- sample(1:L, 1)
    spin <- lattice[i, j]
    neighbors <- lattice[(i %% L) + 1, j] + lattice[(i - 2) %% L + 1, j] +
                 lattice[i, (j %% L) + 1] + lattice[i, (j - 2) %% L + 1]
    delta_E <- 2 * spin * neighbors
    if (delta_E <= 0 || runif(1) < exp(-beta * delta_E)) {
      lattice[i, j] <- -spin
    }
  }
  return(lattice)
}

plot_lattice <- function(lattice, title) {
  library(ggplot2)
  lattice_df <- expand.grid(x = 1:nrow(lattice), y = 1:ncol(lattice))
  lattice_df$spin <- as.vector(lattice)
  ggplot(lattice_df, aes(x, y, fill = factor(spin))) +
    geom_tile(color = "white") +
    scale_fill_manual(values = c("-1" = "blue", "1" = "red")) +
    theme_void() +
    labs(title = title, fill = "Spin")
}

# Example execution
L <- 64  # Lattice size
steps <- c(0, 1, 4, 32, 100, 1000)
T <- 2.5  # Temperature
beta <- 1 / T

# 初始化格點與參數
lattice <- initialize_lattice(L) # 初始隨機格點
plots <- list() # 儲存每個時間步驟的圖

# 生成不同時間步驟的格點圖
for (t in 1:length(steps)) {
  step <- steps[t]
  for (s in 1:step) {
    lattice <- metropolis_step(lattice, beta, L)
  }
  plots[[t]] <- plot_lattice(lattice, paste("Time =", step)) # 將圖儲存到列表
}

# 使用 grid.arrange 將所有圖顯示在同一畫布上
grid.arrange(grobs = plots, ncol = 3)
```



2. Influence of Temperature

The temperature T significantly impacts the behavior of the Ising model: - **Low** T: Spins tend to align, forming large ordered clusters. - **High** T: Spins become more random, resembling a disordered state.

Code for Different Temperatures:

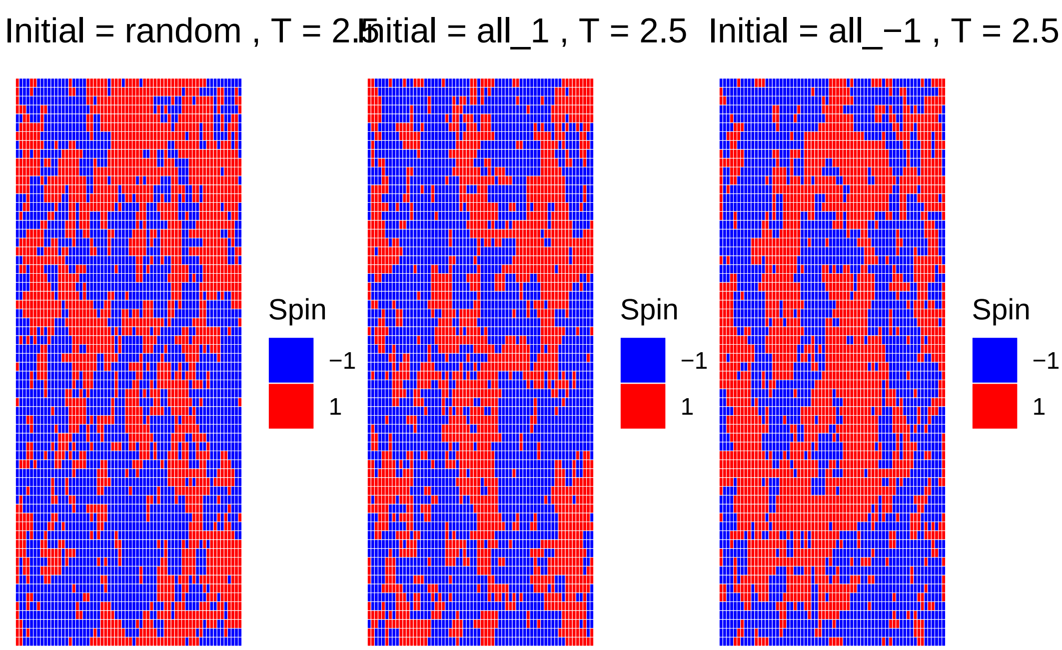


3. Different Initial Conditions

The initial configuration of spins can affect the time to reach equilibrium but not the final equilibrium state (which depends on T).

Code for Different Initial Conditions:

```
initialize_lattice_custom <- function(L, type = "random") {
  if (type == "random") {
    return(matrix(sample(c(-1, 1), L * L, replace = TRUE), L, L))
  } else if (type == "all_1") {
    return(matrix(1, L, L))
  } else if (type == "all_-1") {
    return(matrix(-1, L, L))
  }
}

plot_initial_conditions <- function(L, steps, T, initial_types) {
  plots <- list()
  for (type in initial_types) {
    lattice <- initialize_lattice_custom(L, type)
    beta_T <- 1 / T
    for (step in steps) {
      for (s in 1:step) {
        lattice <- metropolis_step(lattice, beta_T, L)
      }
    }
    plots[[type]] <- plot_lattice(lattice, paste("Initial =", type, ", T =", T))
  }
  do.call(grid.arrange, c(plots, ncol = length(initial_types)))
}

initial_types <- c("random", "all_1", "all_-1")
plot_initial_conditions(L, steps, 2.5, initial_types)
```



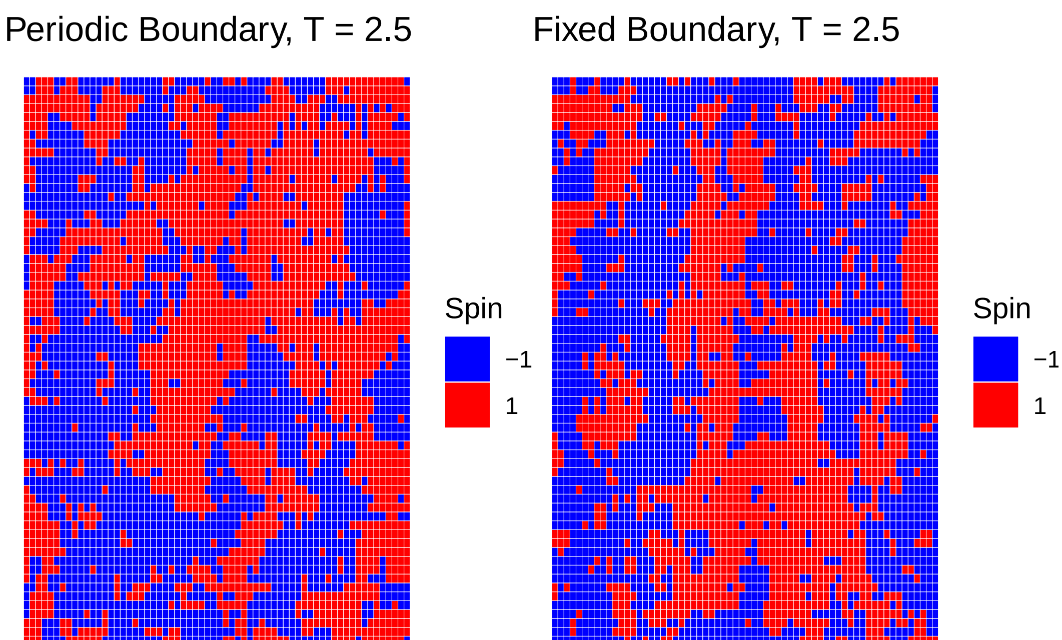
4. Different Boundary Conditions

Boundary conditions determine how spins at the edges interact: 1. **Periodic**: Spins at boundaries wrap around. 2. **Fixed**: Spins at boundaries are fixed to a particular value (e.g., $+1$ or -1).

Code for Boundary Conditions:

Fixed Boundary:

```
initialize_fixed_boundary <- function(L, boundary_spin = 1) {
  lattice <- matrix(sample(c(-1, 1), (L - 2) * (L - 2), replace = TRUE), L - 2, L - 2)
  lattice_full <- matrix(boundary_spin, L, L)
  lattice_full[2:(L-1), 2:(L-1)] <- lattice
  return(lattice_full)
}

plot_boundary_conditions <- function(L, steps, T) {
  periodic_lattice <- initialize_lattice(L)
  fixed_lattice <- initialize_fixed_boundary(L, 1)
  beta_T <- 1 / T

  for (step in steps) {
    for (s in 1:step) {
      periodic_lattice <- metropolis_step(periodic_lattice, beta_T, L)
      fixed_lattice <- metropolis_step(fixed_lattice, beta_T, L)
    }
  }

  p1 <- plot_lattice(periodic_lattice, "Periodic Boundary, T = 2.5")
  p2 <- plot_lattice(fixed_lattice, "Fixed Boundary, T = 2.5")
  grid.arrange(p1, p2, ncol = 2)
}

plot_boundary_conditions(L, steps, 2.5)
```



5. Results

By exploring the influence of temperature, initial conditions, and boundary conditions, the following observations can be made: 1. **Temperature**: - Low T: Ordered clusters form. - High T: Spins are random. 2. **Initial Conditions**: - Affects time to equilibrium but not the final state. 3. **Boundary Conditions**: - Periodic boundaries lead to smoother behavior. - Fixed boundaries influence edge behavior.

This setup allows for a comprehensive study of the 2D Ising model using the Metropolis algorithm.