Reinforcement Learning - Assignment 2

Thalagalage Don Lahiru, Nazanin Baramaki

21st of July 2025

You can find the source code on GitHub: View Source Code on GitHub

1 Part 1

1.1 Estimating the State Value for Grid World

A grid world of 5x5 is considered for the calculation of the state values under:

- A **uniform random policy** (equiprobable in all directions)
- A **discount factor** γ=0.95

1.1.1 Explicitly Solving the Bellman Equation

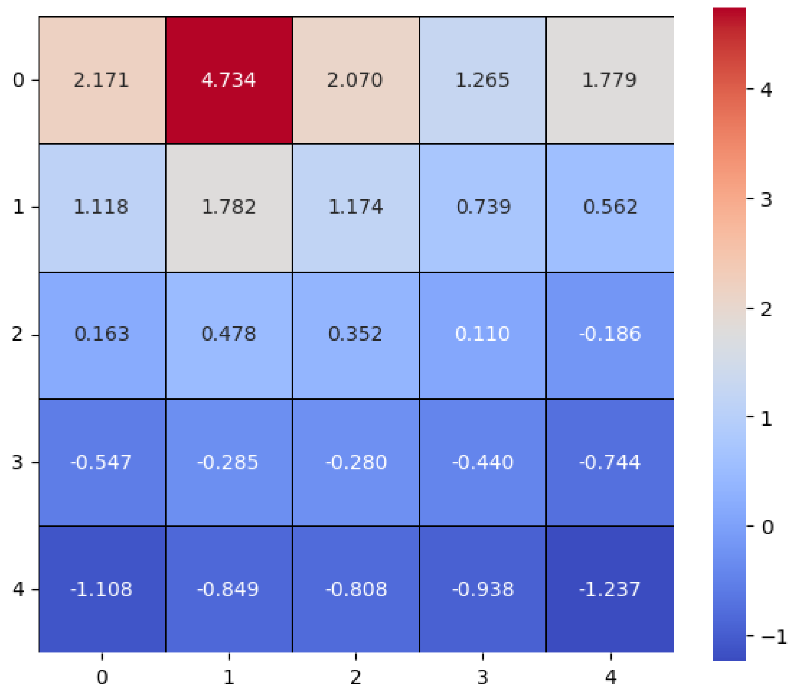

Figure 1: State Values Calculated via Explicit Solution of the Bellman Equation

Figure 1 shows the state values of the grid after explicitly solving the Bellman equation. The value of the maximum state **appears in the top left (0,1)**, where the blue square is located, which justifies the highest reward received from that State. We can also observe that the highest state values are situated around the maximum state value.

1.1.2 Iterative Policy Evaluation (Threshold value = 10^{-4})

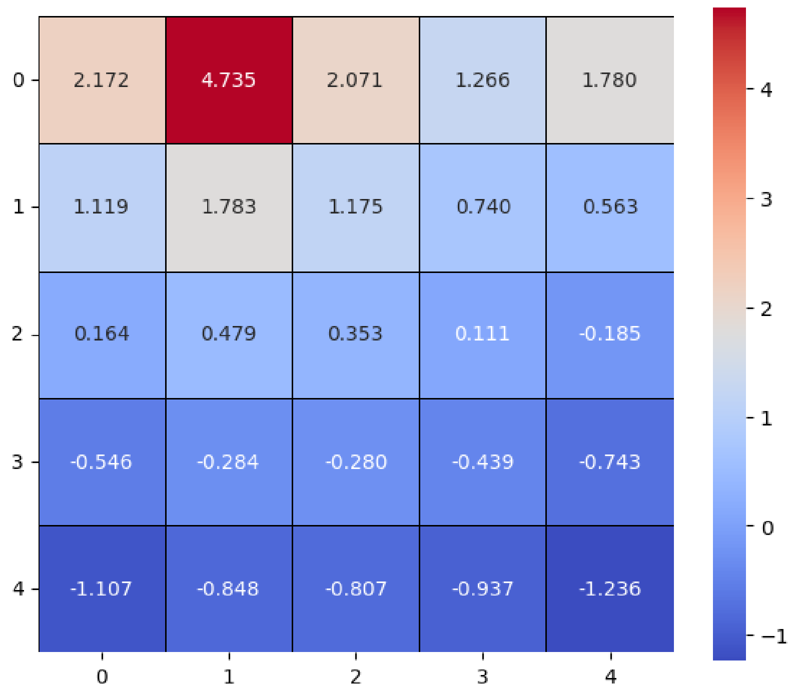

Figure 2: State Values Calculated via Iterative Policy Evaluation

Figure 2 shows the state values of the grid after the iterative evaluation of the policy. The value of the maximum state **appears in the top left (0,1)**, where the blue square is located, which justifies the highest reward received from that State. We can also observe that the highest state values are situated around the maximum state value.

1.1.3 Discussion

The values are very close between the two images, which verifies the numerical consistency between the two methods. The maximum value is located in the top-left region, at **State (0,1)** with values:

- Bellman Equation: **4.734**
- Iterative Policy: **4.735**

These are the **highest value states** in both grid heatmaps.

This result is not surprising due to:

- In this grid world **state (0,1)** has a **high reward of +5**, resulting in driving the **high reward region**.
- Bellman equation function incorporates **future expected rewards**. Because of that, states closer to the **high reward state get higher values**.
- Due to the **uniform random policy**, expected returns **fluctuate randomly**, so naturally, states closer to the higher reward regions accumulate higher expected returns.

1.2 Determining the Optimal Policy for Grid World

1.2.1 Explicitly Solving the Bellman Equation

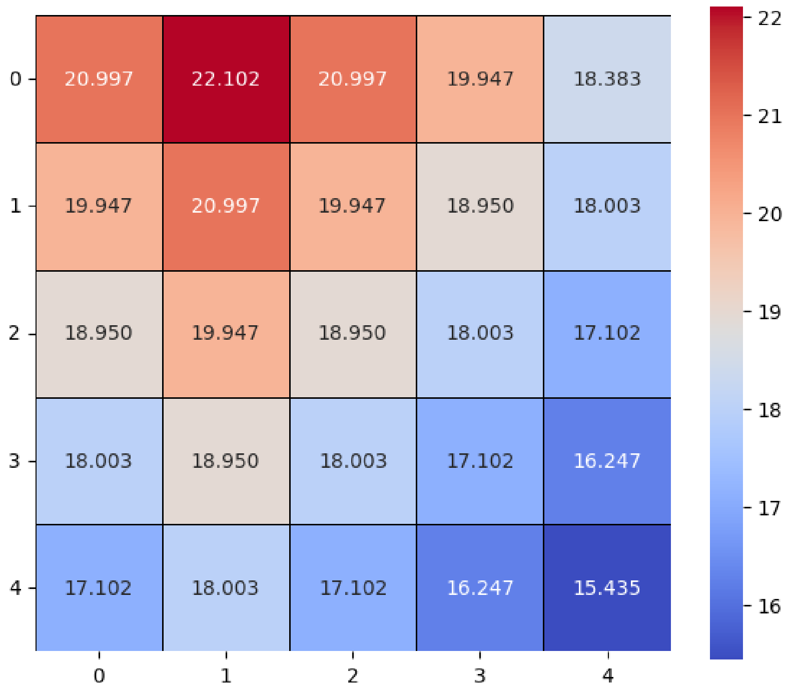

Figure 3: State Values Calculated via Explicit Solution of the Bellman Equation

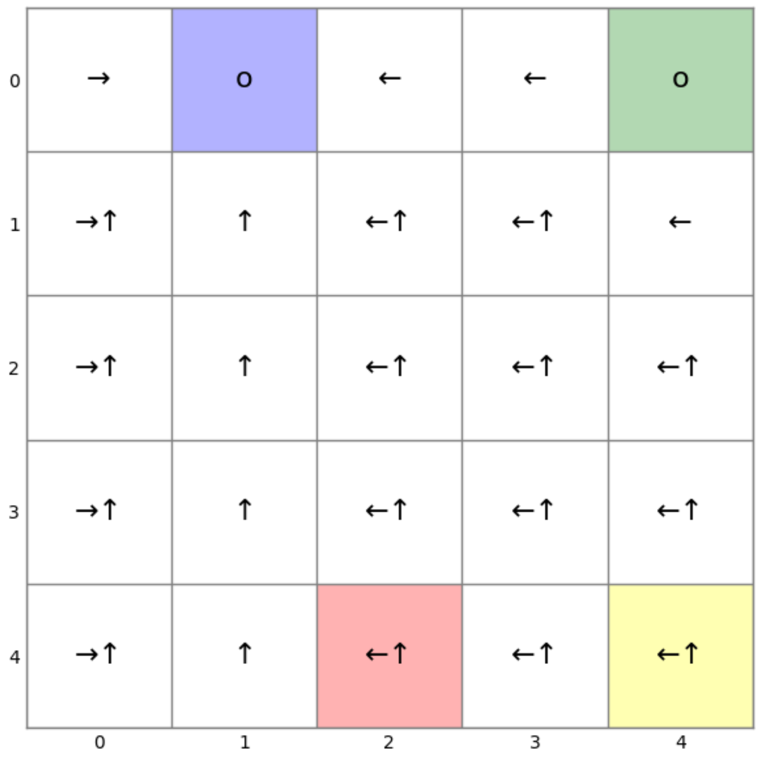

Figure 4: State Values Calculated via Explicit Solution of the Bellman Equation

Figure 3 and 4 show each State's value function and policy, respectively.

- The heatmap demonstrates the value function maximizing at State (0,1) with value ≈ 22.102.
- The arrows in the policy plot illustrate the direction that maximizes value - approach is consistent with the greedy policy derived from the optimal value function.

1.2.2 Iterative Policy Evaluation (Threshold value = 10^{-4})

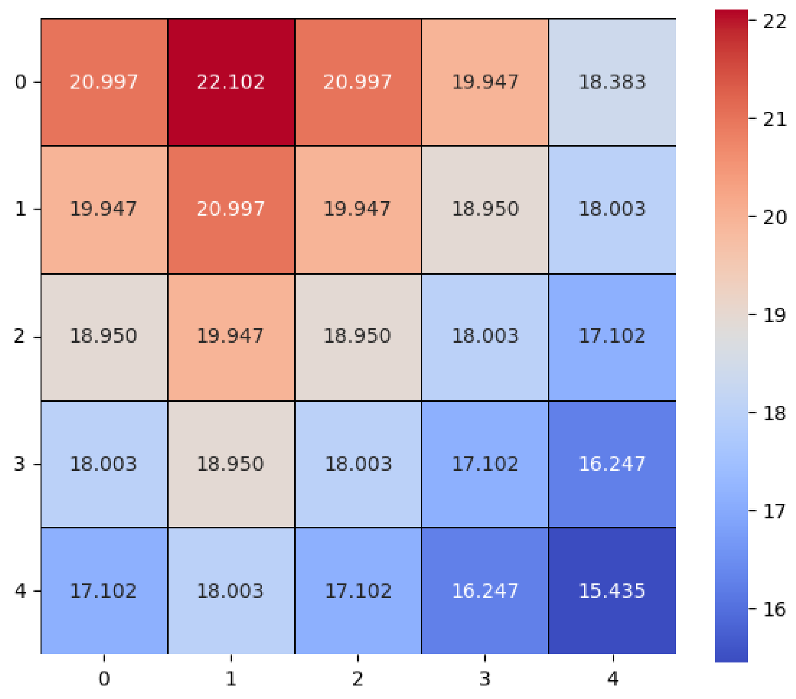

Figure 5: State Values Calculated via Explicit Solution of the Iterative Policy Evaluation

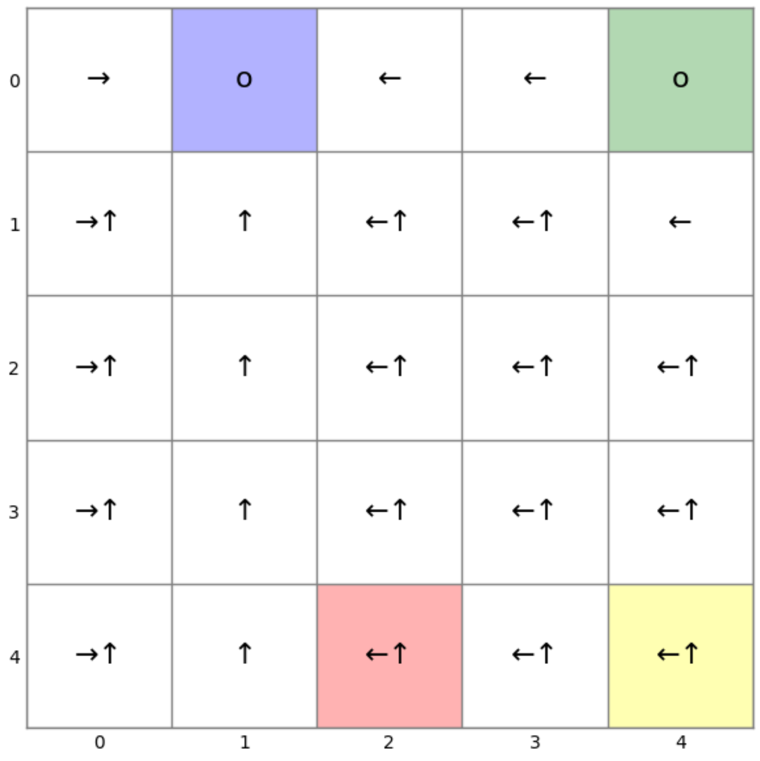

Figure 6: State Values Calculated via Explicit Solution of the Iterative Policy Evaluation

Figures 5 and 6 show each State's value function and policy, respectively.

- Final value function equivalent to that from Bellman optimality, confirming convergence.
- Optimal policy arrows are equal to the Bellman optimality policy, confirming the convergence.

1.2.3 Policy Improvement with Value Iteration(Threshold value = 10^{-4})

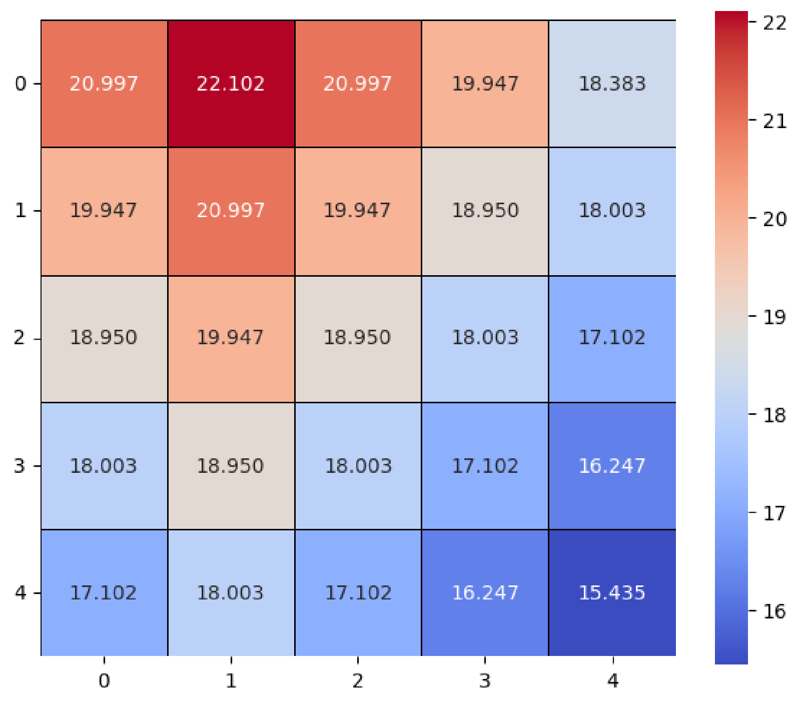

Figure 7: State Values Calculated via Policy Improvement with Value Iteration

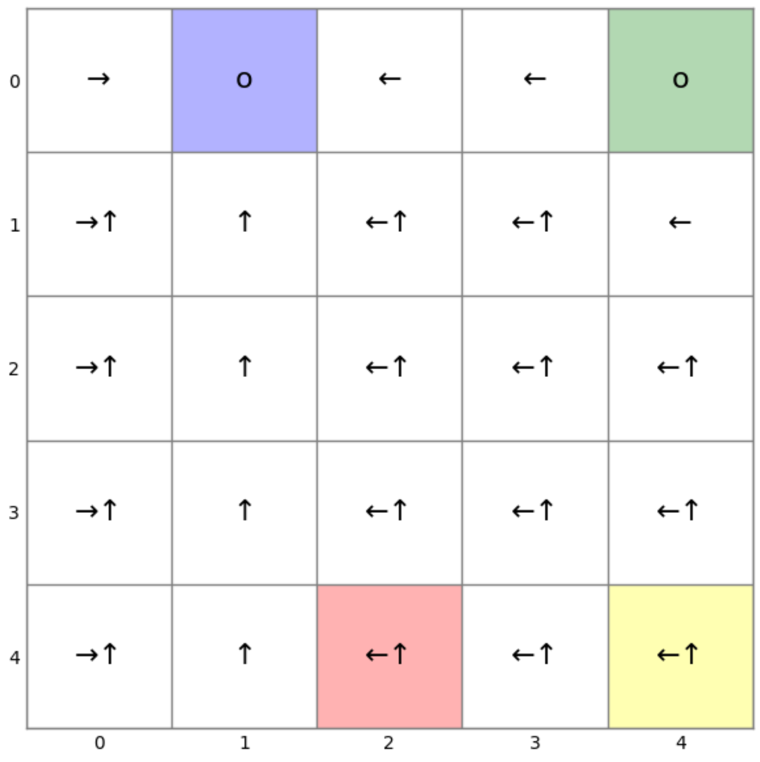

Figure 8: State Values Calculated via Policy Improvement with Value Iteration

Figure 7 and 8 show the value function and policy for each State, respectively.

- Value function matches Bellman/Iteratively Policy results.
- Policy is again identical to Bellman and Policy Iteration, which confirms the convergence.

1.2.4 Conclusion

All methods demonstrate that **state(0,1)** has the highest value $\approx$ **22.102**. All the related actions will likely move **towards the goal state**.

2 Part 2

2.1 Monte Carlo method to learn Optimal Policy for Grid World

2.1.1 Exploring Starts

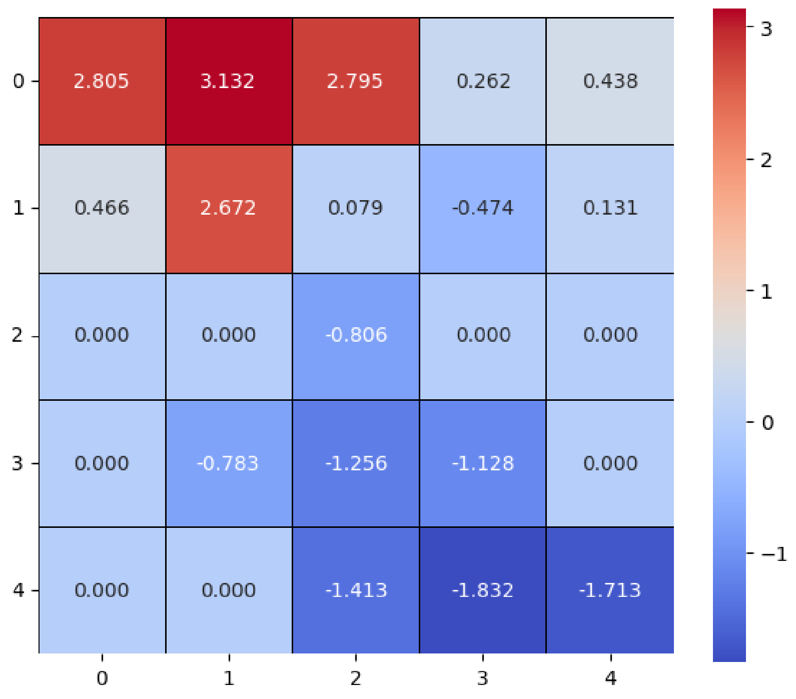

Figure 9: State Values Calculated via Exploring Starts

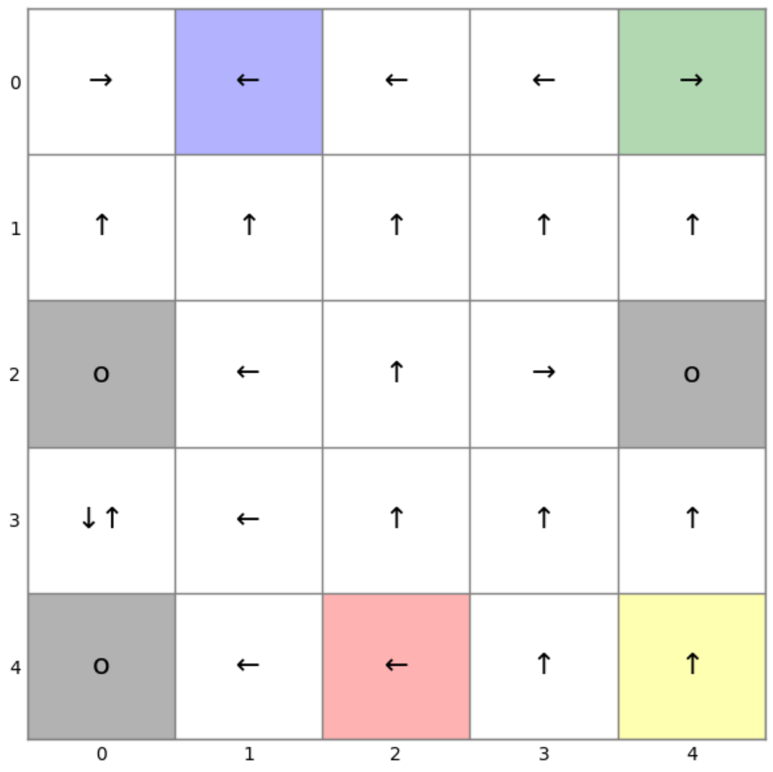

Figure 10: State Policy Calculated via Exploring Starts

Value function summary:

- Highest value ≈ **3.132** at (0,1).
- Higher values on the top and values drop to negative near the bottom right corner.
- Terminal states (grey) correctly show 0. (no future rewards)

Policy summary:

- Deterministic and follows a greedy policy towards the goal. (green)
- Analyzing the arrows shows us that up arrows dominate and towards high value states while avoiding terminal states, which indicates the policy is well aligned with value function gradients.

2.1.2 ϵ-soft Approach

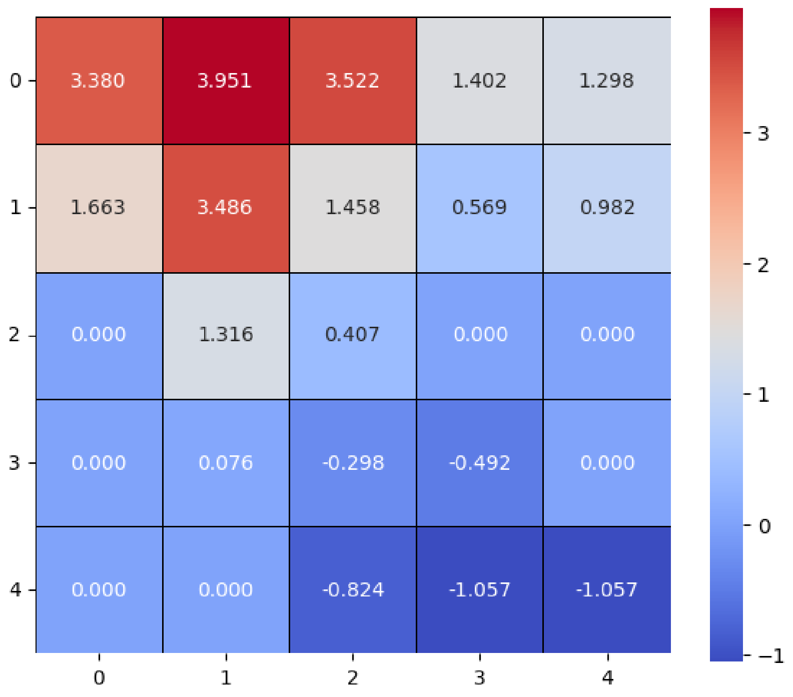

Figure 11: State Values Calculated via Monte Carlo method with ϵ-soft (ϵ = 0.85)

Value function summary:

- Higher peak value ≈ **3.951** at (0,1).
- Similar to exploring starts, but slightly improved top row values.
- Negative values are smaller in magnitude. (e.g., -1.057 vs -1.832)

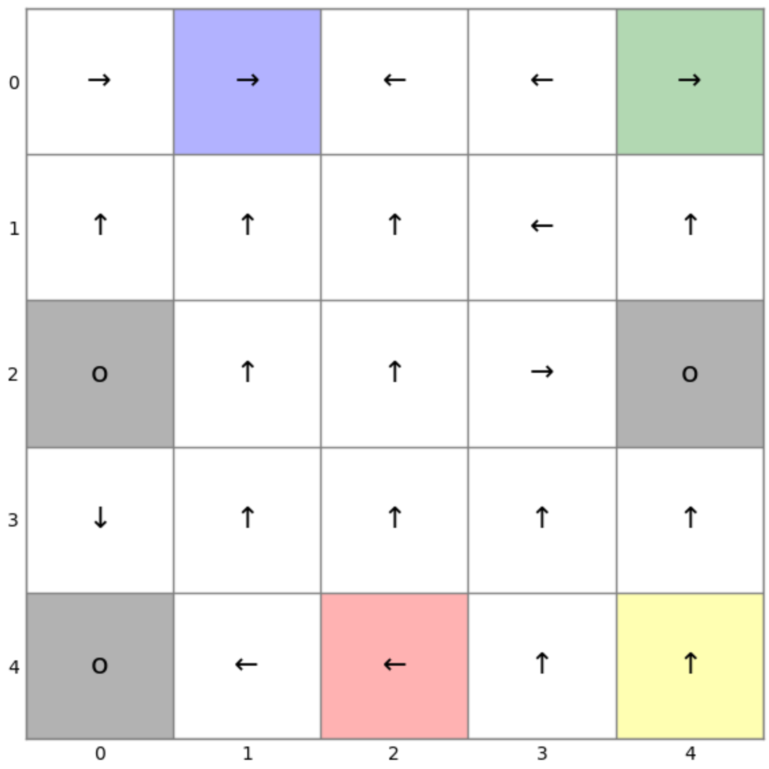

Figure 12: State Policy Calculated via Monte Carlo method with ϵ-soft ($\epsilon = 0.85$)

Policy summary:

- Arrows are similar to the previous method, showing consistent convergence.
- Policy avoids terminal states and guides the agent towards the top-right, less noisy.

Different ϵ values were used and tested to analyze how ϵ-soft algorithm behaves:

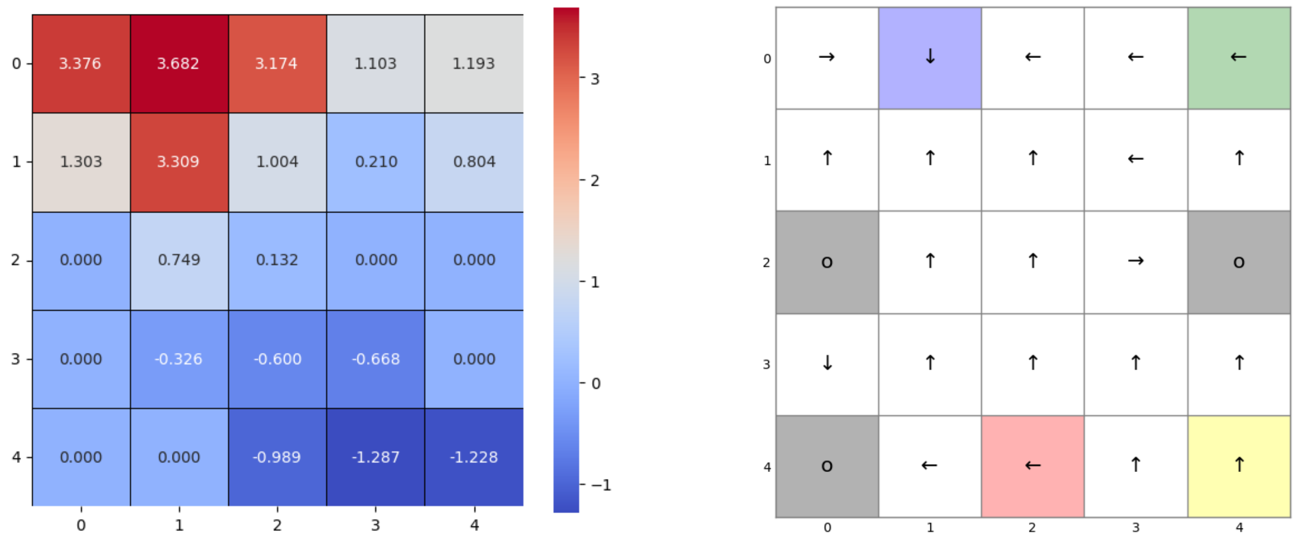

Figure 13: State Values and Policy Estimated via Monte Carlo method with ϵ-soft ($\epsilon = 0.9$)

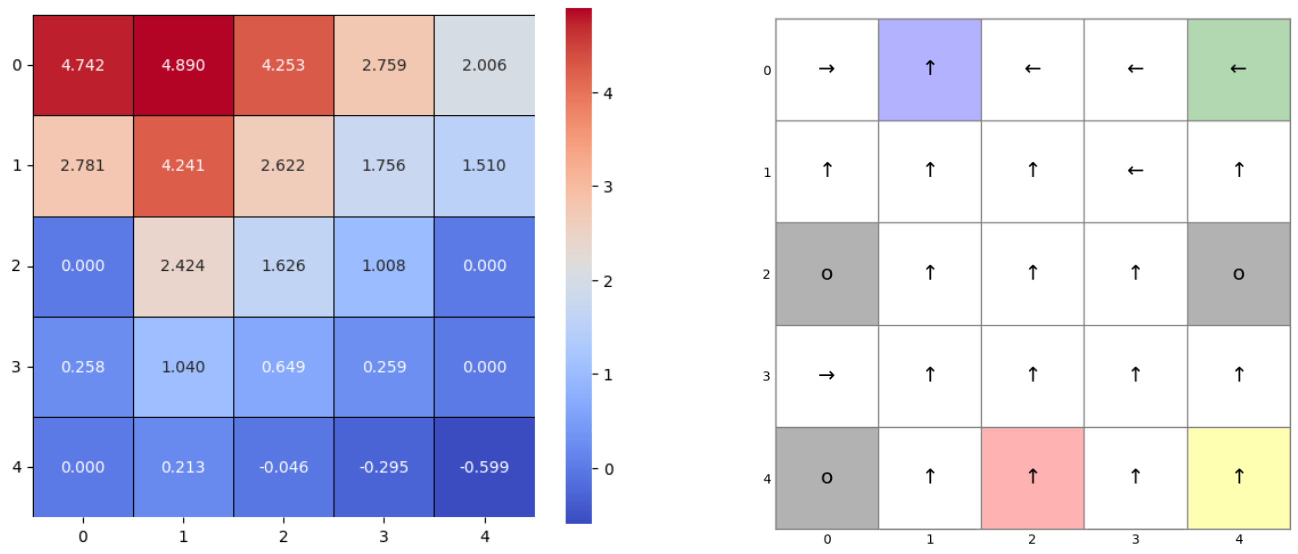

Figure 14: State Values and Policy Estimated via Monte Carlo method with ϵ-soft ($\epsilon = 0.7$)

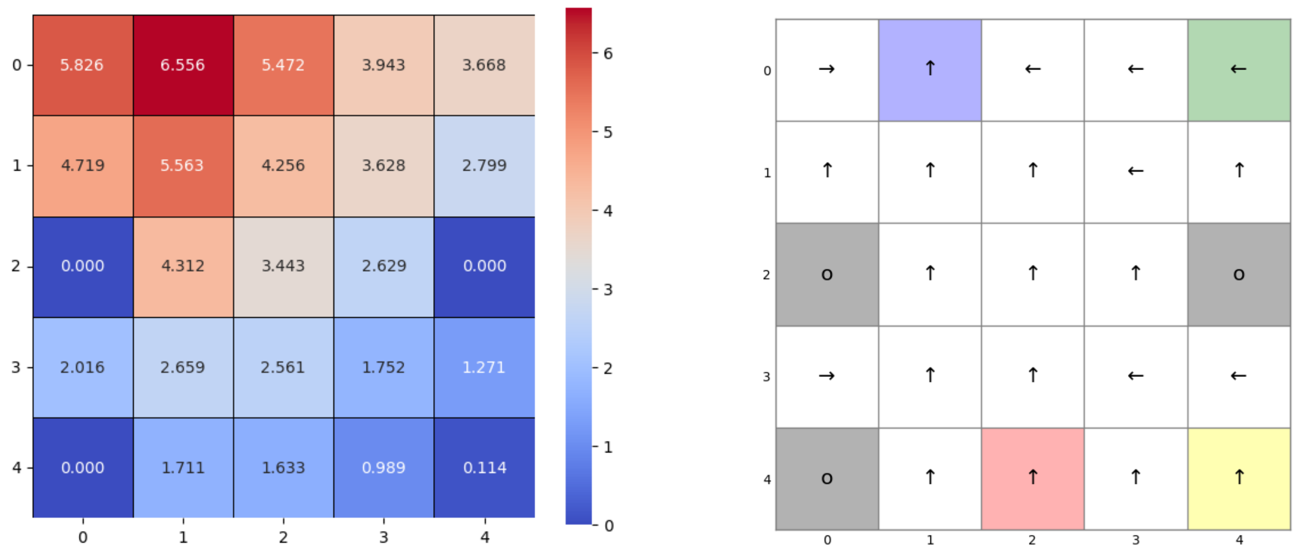

Figure 15: State Values and Policy Estimated via Monte Carlo method with ϵ-soft ($\epsilon = 0.5$)

As the ϵ value increases in the ϵ-soft policy, the policy becomes more exploratory, allowing higher probabilities to suboptimal actions. This results in **slower convergence**, and in general, **value estimates are lower**, as you can see from the Figure 13 ($\epsilon = \mathbf{0.9}$). In contrast, **smaller ϵ values ($\epsilon = 0.5$)** allow more exploration, and **higher value estimates** with a more optimal-looking policy. However, the **right amount of exploration** is needed to avoid **local optima** and to make sure **all the state and action** pairs are visited.

2.1.3 Conclusion

1. **Monte Carlo with exploring starts** results in a near-optimal policy by covering all states and actions via varying starting conditions.
2. **Monte Carlo with ϵ-soft (no exploring starts)** also learns the best policy by exploration via randomness. It produces slightly better value estimates and smoother convergence than the previous method.

2.2 Equiprobable Behaviour Policy to learn the Optimal Policy

The agent learns the optimal policy using off-policy Monte Carlo Control with importance sampling, where:

- Behaviour policy is uniformly random (equiprobable moves).
- Returns and importance sampling are used to improve the target policy greedily.
- Exact importance weights between behaviour and target policies can be computed, since the transition dynamics are known.

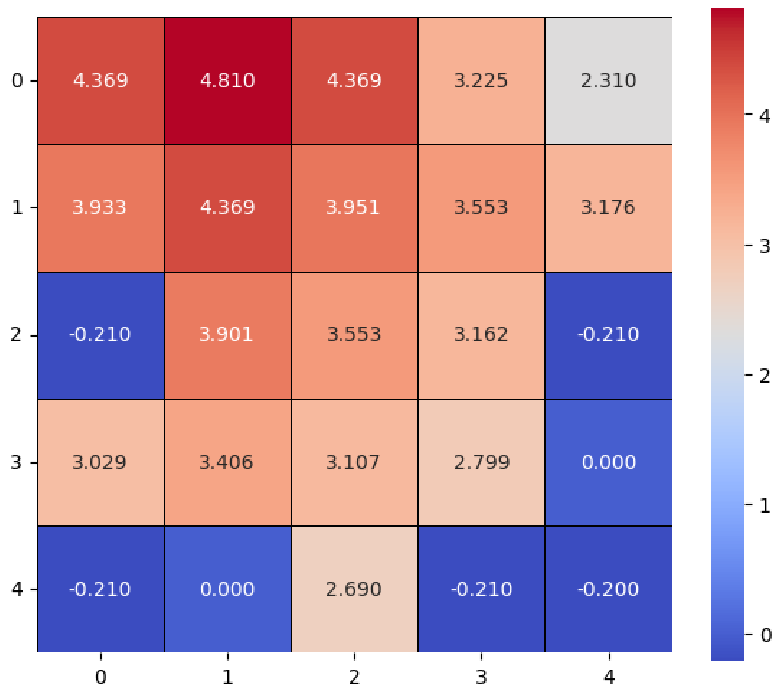

Figure 16: State Values Calculated via Equiprobable Behaviour Policy

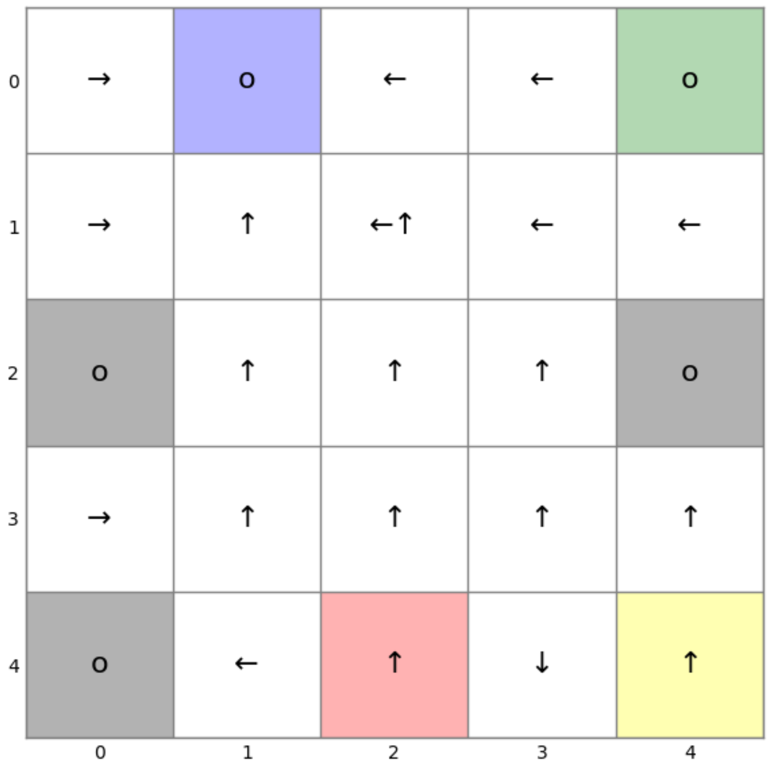

Figure 17: State Policy Calculated via Equiprobable Behaviour Policy

Figure 16 and 17 show the value function and policy for each State, respectively.

Value Function Analysis:

- Highest value is at State (0,1), with value 4.810 and the rest of the higher value states are located on the top rows (e.g., (0,0), (0,2), (1,1)).

Policy Analysis:

- The learned policy guides the agent toward higher value states and avoids terminal states.
- Policy predominantly points to the top and sometimes to the right and left based on the proximity of the high-value region.
- The policy can be considered nearly deterministic, due to the convergence to the optimal policy under off-policy learning.

2.2.1 Conclusion

Considering the above value function and policy, the agent successfully avoids negative rewards and terminal states and reaches goal regions effectively.