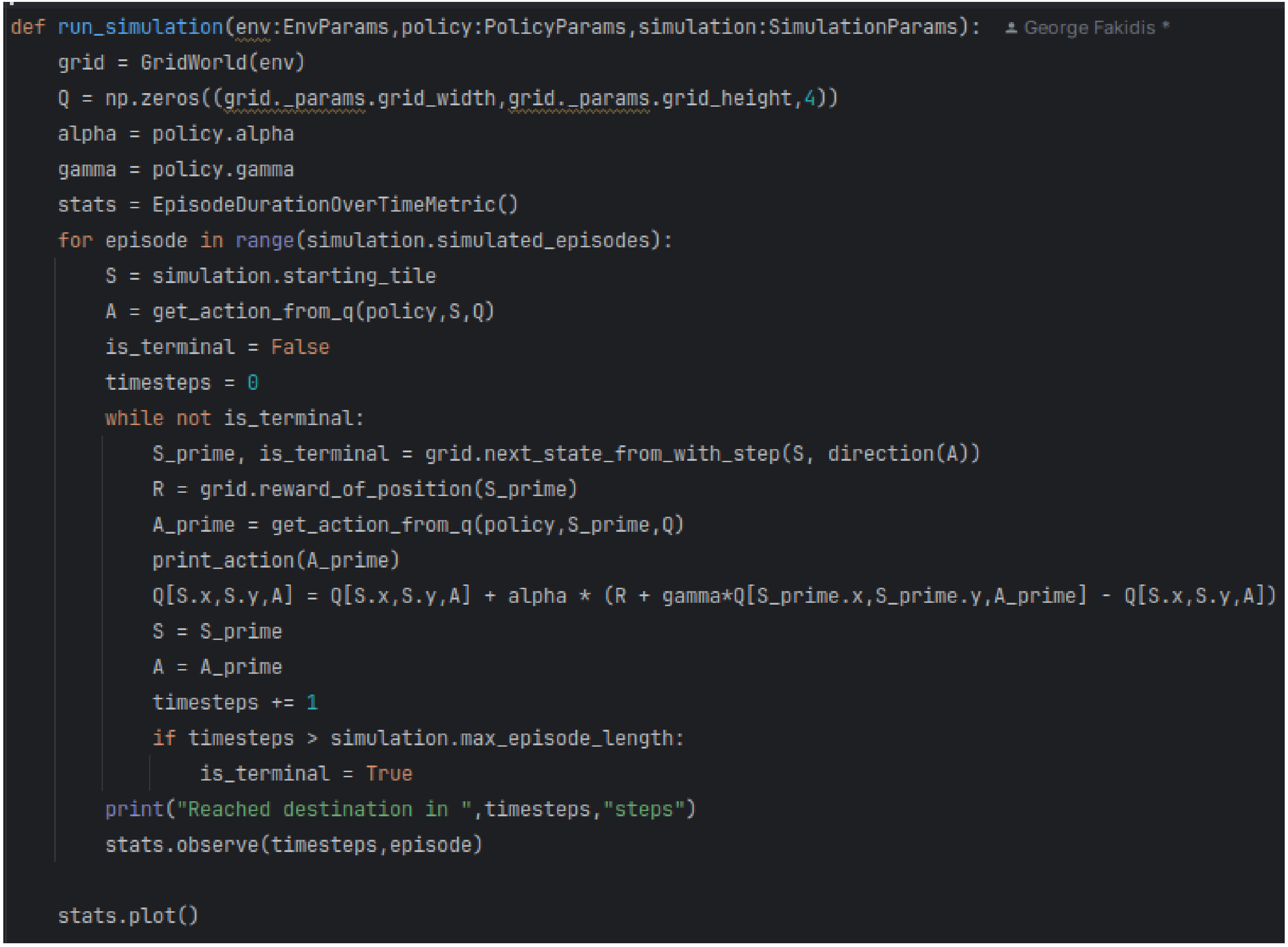
Solving the Windy GridWorld with SARSA

George Fakidis, Athens University of Economics and Business

January 12, 2025

1 Code

The full code can be found in My Github.

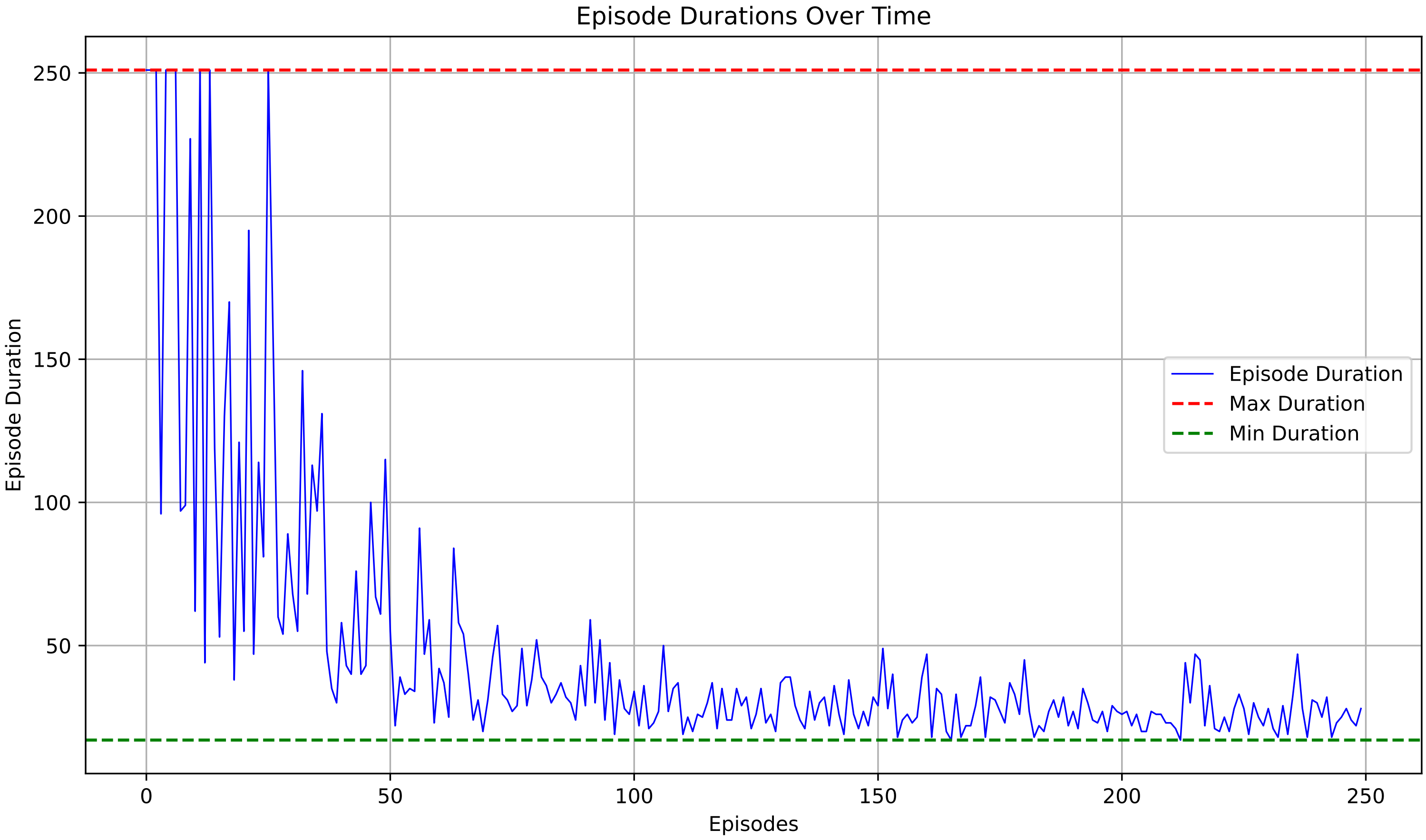


```python
def run_simulation(env:EnvParams,policy:PolicyParams,simulation:SimulationParams):
    grid = GridWorld(env)
    Q = np.zeros((grid._params.grid_width,grid._params.grid_height,4))
    alpha = policy.alpha
    gamma = policy.gamma
    stats = EpisodeDurationOverTimeMetric()
    for episode in range(simulation.simulated_episodes):
        S = simulation.starting_tile
        A = get_action_from_q(policy,S,Q)
        is_terminal = False
        timesteps = 0
        while not is_terminal:
            S_prime, is_terminal = grid.next_state_from_with_step(S, direction(A))
            R = grid.reward_of_position(S_prime)
            A_prime = get_action_from_q(policy,S_prime,Q)
            print_action(A_prime)
            Q[S.x,S.y,A] = Q[S.x,S.y,A] + alpha * (R + gamma*Q[S_prime.x,S_prime.y,A_prime] - Q[S.x,S.y,A])
            S = S_prime
            A = A_prime
            timesteps += 1
            if timesteps > simulation.max_episode_length:
                is_terminal = True
        print("Reached destination in ",timesteps,"steps")
        stats.observe(timesteps,episode)

    stats.plot()
```

Implementation details regarding the GridWorld and the policy were left out intentionally in order to focus on the SARSA algorithm.

2 Charts



This is how the algorithm behaves, the main thing to notice here is that episodes last less as time progresses, meaning our algorithm actually learns how to navigate the windy landscape.