The idea of Policy Iteration algorithm can be applied using both Dynamic Programming and Monte Carlo Simulation, although both can be proved to have policy improvement at each iteration, the algorithm that has such guarantee is not exactly the same, so are the proofs.

The document is a comparison of the two algorithms used and their corresponding proof, in the end we show why the PI algorithm used in the dynamic programming case would fail in the Monte Carlo case.

Note that Alg 2 has multiple variants, including non-stationary ϵ(an example is **GLIE**:Greedy in the Limit with Infinite Exploration), incremental implementation, fix step size α, first-visit/every-visit MC etc. Here we only show one of them.

1. The basic policy iteration algorithm when given the MDP is as Alg 1

Algorithm 1: Policy Iteration

Input: Markov Decision Process $\langle S, A, P, \mathcal{R}, \gamma \rangle$
Output: V^*, π^*
Initialisation : $V(s) \in \mathbb{R}, \pi(s) \in A(s).$ arbitrarily, $\forall s \in S$
policy-stable $\leftarrow$ `False`
while *policy-stable* $=$ `False` **do**
  Iterative **Policy Evaluation**: $V(s) \leftarrow V^\pi(s)$
  $\Delta \leftarrow \theta$
  while $\Delta \geq \theta$ **do**
    for $s \in S$ **do**
      $v \leftarrow V(s)$
      $V(s) \leftarrow \sum_{a \in A} \pi(a \mid s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V(s') \right)$
      $\Delta \leftarrow \max(\Delta, \mid v - V(s) \mid)$
    end for
  end while
  Policy Improvement: $\pi \leftarrow \texttt{greedy}(V)$
  policy-stable $\leftarrow$ `False`
  for $s \in S$ **do**
    previous-action $\leftarrow \pi(s)$
    $\pi(s) \leftarrow \operatorname{argmax}_a \sum R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V(s')$
    if *previous-action* $\neq \pi(s)$ **then**
      policy-stable $\leftarrow$ `False`
    end if
  end for
end while
return V, π

The proof of policy improvement is as follows:

Proof. Denote the previous policy as π and the updated policy as π'. Note that according to Algorithm 1, in every policy improvement step, we have $\pi'(s) \leftarrow \text{argmax}_a q_\pi(s,a)$.

$$\begin{aligned}
v_\pi(s) &= q_\pi(s, \pi(s)) \\
&\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right] \\
&\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid s_t = s\right] \\
&\vdots \\
&\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s\right] \\
&= v_{\pi'}(s)
\end{aligned}$$

■

2. The generalized policy iteration algorithm used for Monte-Carlo control is as Alg 1. Note that this is an **on-policy** algorithm which attempts to evaluate/improve the policy that's used to make decisions.

   It's important to note the two differences of this algorithm compared with Algorithm 1:

   - we're now using action-value function instead of state-value function. This is because we don't get access to the dynamics(transition and reward) of the MDP anymore. The Bellman Optimality Equation for $V(s)$ requires R and P to perform the one step "look-ahead".
   - we're now using $\epsilon-$greedy policy improvement instead of greedy. This is for the consideration of exploration. Otherwise, if at state s there exists a action a_1 that in expectation has high reward but when we first visit the state we get 0 reward, at the same time there's another action a_2 which always gives us positive reward 1, the greedy algorithm will thus always select a_2.

   First we state the $\epsilon-$greedy policy improvement theorem:

   Theorem 1. *For any ϵ-greedy policy π, the ϵ-greedy policy π' with respect to q_π is an improvement, $v_{\pi'}(s) \geq v_\pi(s)$.*

   The proof is as follows:

   Proof. Denote the previous policy as π and the updated policy as π'. Note that π can be any $\epsilon-$greedy policy. And we observe the following equality:

$$\sum_a \frac{\pi(a \mid s) - \frac{\epsilon}{|A|}}{1-\epsilon} = \frac{1}{1-\epsilon} - \frac{\epsilon}{1-\epsilon} = 1$$

Algorithm 2: Generalized Policy Iteration for On-policy Every-visit Monte-Carlo Control

Input: State space S, action space A, environment interface $(s', r) \leftarrow f(s, a)$, discount factor γ, exploration probability ϵ
Output: π^*
Initialisation : $Q(s, a) \in \mathbb{R}, \forall s \in S, a \in A.$ arbitrarily, $N_{episode} \leftarrow 0$

while want new episode **do**
    $N_episode \leftarrow N_episode + 1$
    Sample an episode following π: $\{S_1, A_1, R_1, S_2, \cdots, S_{T-1}, A_{T-1}, R_{T-1}, S_T\}$
    Monte-Carlo **Policy Evaluation** with action-value function:
    for $t \in [1, \cdots, T)$ **do**
        $N_{(S_t, A_t)} \leftarrow N_{(S_t, A_t)} + 1$
        $G_t \leftarrow \sum_t^{T-1} R_t$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{G_t - Q(S_t, A_t)}{N_{(S_t, A_t)}}$
    end for
    $\epsilon-$greedy **Policy Improvement**:
    for $s \in S$ **do**
        $A^* \leftarrow \text{argmax}_a Q(s, a)$
        for $a \in A$ **do**
            if $a = A^*$ **then**
                $\pi(a|s) \leftarrow 1 - \epsilon + \frac{\epsilon}{|A|}$
            else
                $\pi(a|s) \leftarrow \frac{\epsilon}{|A|}$
            end if
        end for
    end for
end while
return π

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_a \pi'(a \mid s) q_\pi(s, a) \\
&= \frac{\epsilon}{\mid A \mid} \sum_a q_\pi(s, a) + (1 - \epsilon) \max_a q_\pi(s, a) \\
&\geq \frac{\epsilon}{\mid A \mid} \sum_a q_\pi(s, a) + \sum_a \frac{\pi(a \mid s) - \frac{\epsilon}{|A|}}{1 - \epsilon} q_\pi(s, a) \\
&= \sum_a \pi(a \mid s) q_\pi(s, a) \\
&= v_\pi(s)
\end{aligned}
$$

With the above inequality, we simply use the same trick to get $v_{\pi'}(s) \geq q_\pi(s, \pi'(s))$ and thus $v_{\pi'}(s) \geq v_\pi(s)$.

$\blacksquare$