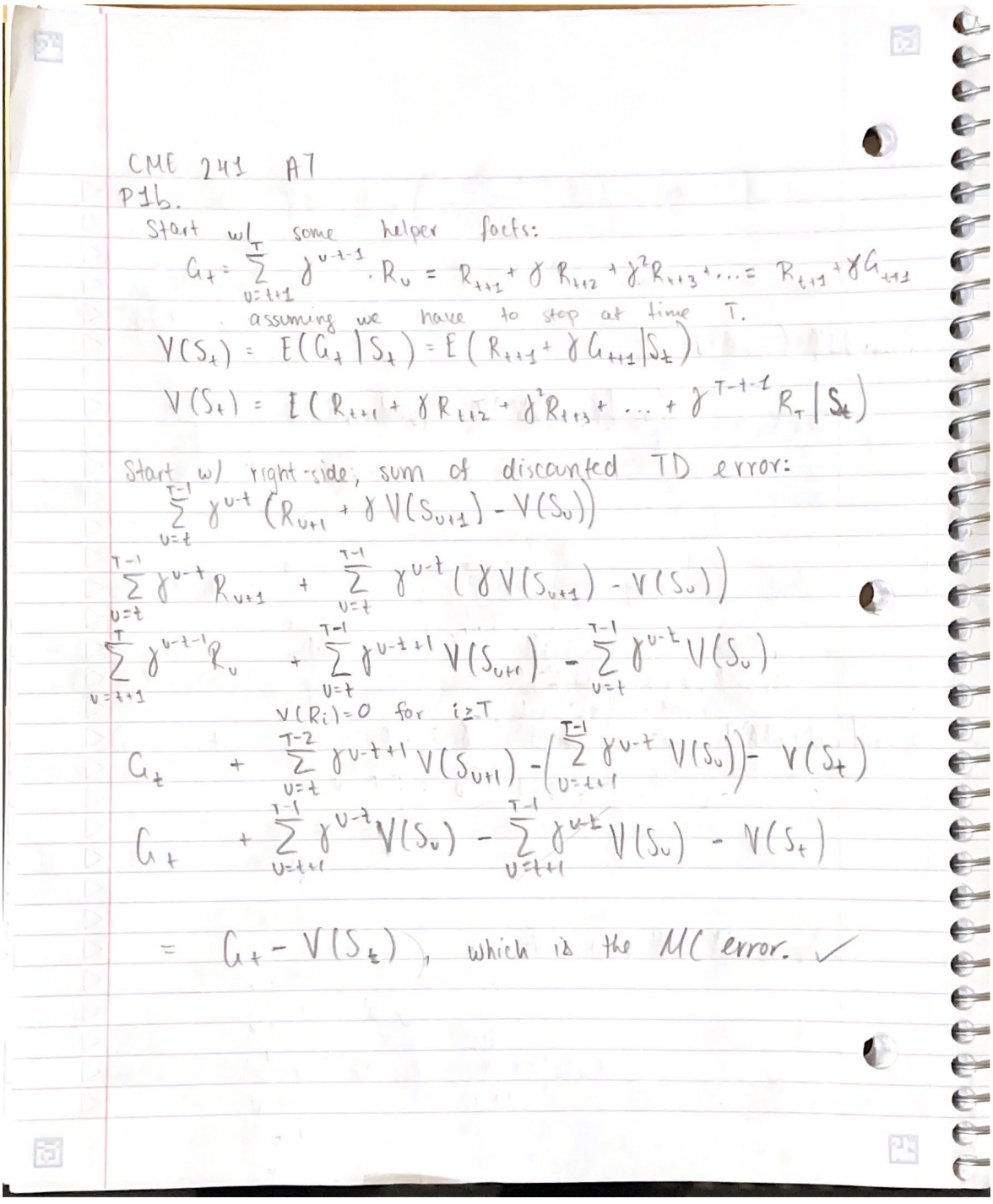
**Question 0, Homework 6, CME241**

Link to code: https://github.com/nabilah13/RL-book/tree/master/assignment6

Done in collaboration with Spencer Siegel, Johannes Fuest

(a) Done in code. https://github.com/nabilah13/RL-book/blob/master/assignment6/a6p1.py



CME 241 A7

P1b.

Start w/ some helper facts:

$$G_t = \sum_{u=t+1}^{T} \gamma^{u-t-1} \cdot R_u = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = R_{t+1} + \gamma G_{t+1}$$

assuming we have to stop at time T.

$$V(S_t) = E(G_t | S_t) = E(R_{t+1} + \gamma G_{t+1} | S_t)$$

$$V(S_t) = E(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T | S_t)$$

Start w/ right-side, sum of discounted TD error:

$$\sum_{u=t}^{T-1} \gamma^{u-t} (R_{u+1} + \gamma V(S_{u+1}) - V(S_u))$$

$$\sum_{u=t}^{T-1} \gamma^{u-t} R_{u+1} + \sum_{u=t}^{T-1} \gamma^{u-t} (\gamma V(S_{u+1}) - V(S_u))$$

$$\sum_{u=t+1}^{T} \gamma^{u-t-1} R_u + \sum_{u=t}^{T-1} \gamma^{u-t+1} V(S_{u+1}) - \sum_{u=t}^{T-1} \gamma^{u-t} V(S_u)$$

$V(R_i) = 0$ for $i \geq T$

$$G_t + \sum_{u=t}^{T-2} \gamma^{u-t+1} V(S_{u+1}) - \left( \sum_{u=t+1}^{T-1} \gamma^{u-t} V(S_u) \right) - V(S_t)$$

$$G_t + \sum_{u=t+1}^{T-1} \gamma^{u-t} V(S_u) - \sum_{u=t+1}^{T-1} \gamma^{u-t} V(S_u) - V(S_t)$$

$= G_t - V(S_t)$, which is the MC error. ✓

(b)

(c) I created a simple MarkovRewardProcess instance where the states are the integers 0, 1, 2, ..., 24. the reward function is $r(s) = 20 - |s - 12|$, and the state transition function is that given you are at state s, you have a uniform transition probability between max(0,s-5) and min(24,s+4).

We can see that all the algorithms give us similar value function solutions. Additionally, in the plots, we can see that the TD(lambda) algorithm has convergence that is in between MC and TD, as we would expect, as TD(lambda) is closer to MC as lambda grows larger, closer to TD as it becomes smaller. There is some noise in the plots coming from the traces being regenerated in between each algorithm being run, and also there is a difference in the learning rate schedules being used between the two plots.

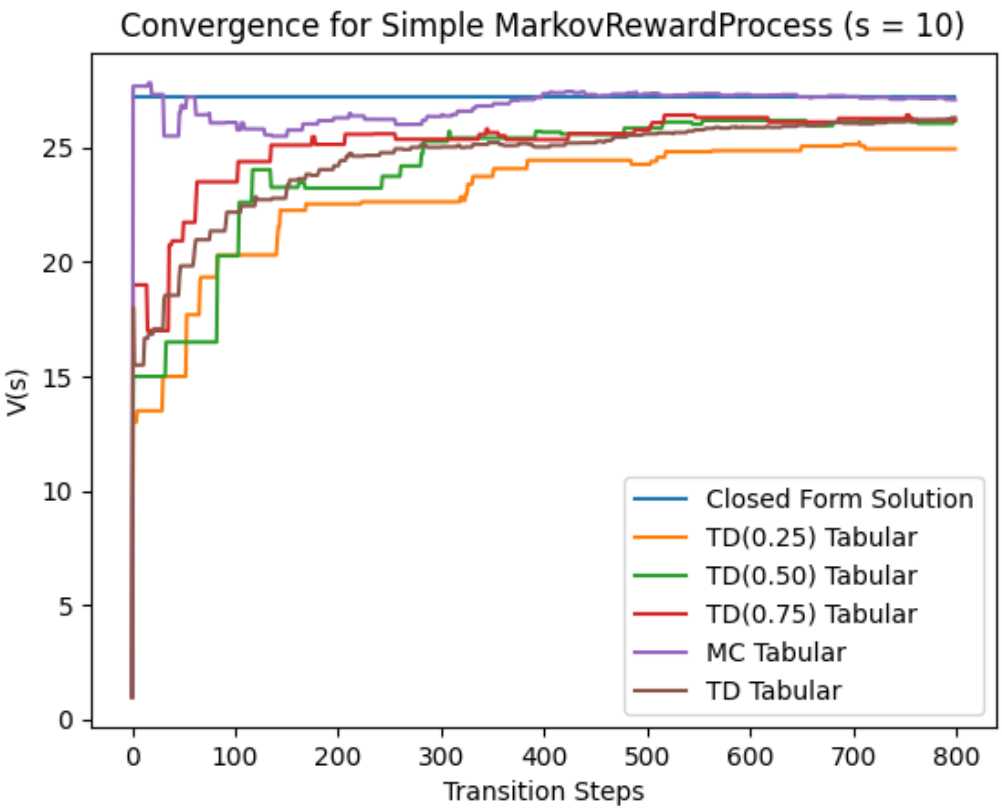
Solution from dynamic programming:
17.56484649 18.20629624 18.88660933 19.59125397 20.31157504 21.03386258
22.42121963 23.80941618 25.18535885 26.33612628 27.24877274 27.91120999
28.28517605 28.36043969 28.12773654 27.59479387 26.77037095 25.69100734
24.36777995 23.01255967 21.63819964 20.92711987 20.20449131 19.47993251
18.76873593

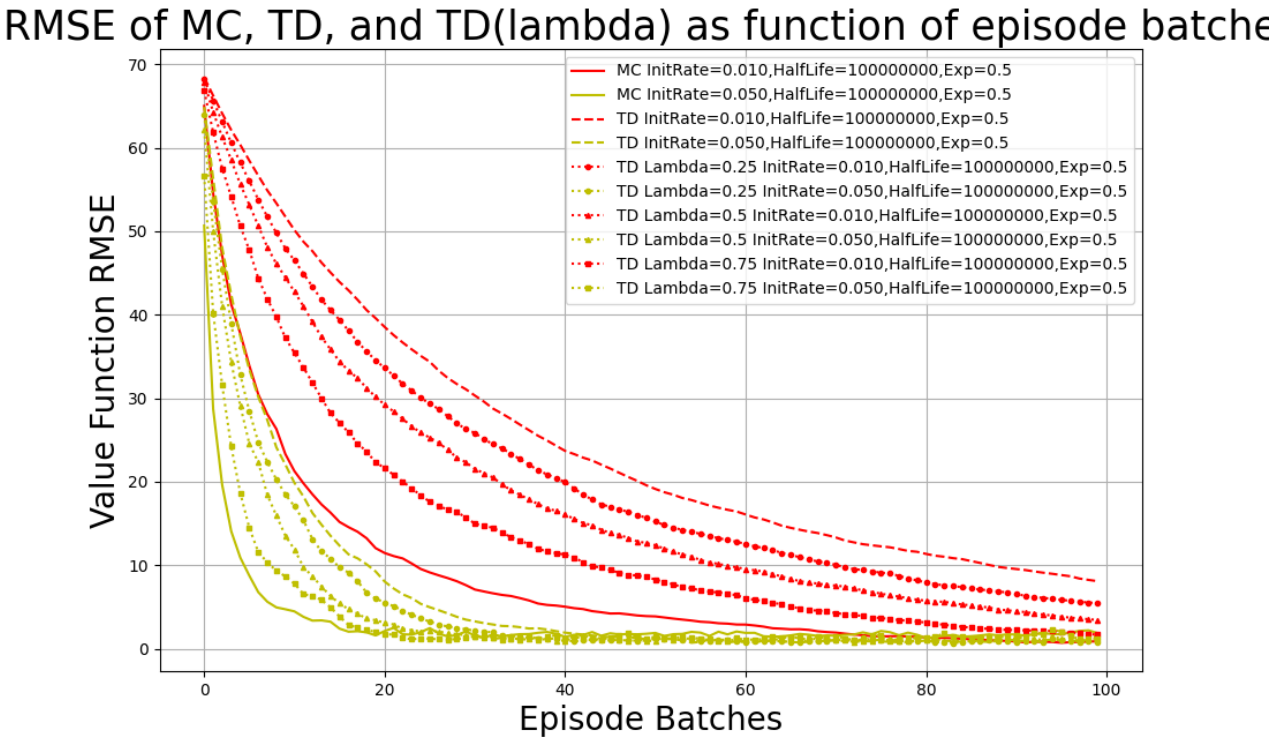
Solution from closed form solution:
17.56484649 18.20629624 18.88660933 19.59125397 20.31157504 21.03386258
22.42121963 23.80941618 25.18535885 26.33612628 27.24877274 27.91120999
28.28517605 28.36043969 28.12773654 27.59479387 26.77037095 25.69100734
24.36777995 23.01255967 21.63819964 20.92711987 20.20449131 19.47993251
18.76873593

Solution from TD(lambda=0.5) tabular:
17.28309199 17.97598709 18.61062069 19.34062414 19.99899739 20.81902895
22.179544 23.74891901 24.93654261 26.09906054 27.10691138 27.54444183
27.66506765 28.10692774 27.71415008 26.95356027 27.03275256 25.1453967
23.78249866 22.27986133 21.16618582 20.60167276 19.75620449 18.82030213
18.78518803

Solution from MC tabular:
17.6732681 18.17289215 18.78790158 19.36419613 20.29397846 20.97896699
22.29827606 23.59707604 24.80189398 26.38140898 27.29246998 27.95134751
28.12813969 28.27115615 28.14947897 27.36591445 26.38078699 25.82815145
24.36348317 23.30405375 21.72702174 20.85253007 19.39030567 19.74940908
19.52966861

Solution from TD tabular:
17.2392795 18.02363201 18.73152336 19.40311267 20.08403805 21.02737611
22.05726605 23.57617579 25.0450957 25.80410718 27.12022409 27.57414538
28.10735521 28.09819198 27.79441575 27.09194452 26.82028447 24.83765085
23.77944133 23.32758922 21.64970555 20.9443741 19.88505938 19.12511755
17.77218264


Convergence for Simple MarkovRewardProcess (s = 10)
V(s)
Transition Steps
Closed Form Solution
TD(0.25) Tabular
TD(0.50) Tabular
TD(0.75) Tabular
MC Tabular
TD Tabular


RMSE of MC, TD, and TD(lambda) as function of episode batche
Value Function RMSE
Episode Batches
MC InitRate=0.010,HalfLife=100000000,Exp=0.5
MC InitRate=0.050,HalfLife=100000000,Exp=0.5
TD InitRate=0.010,HalfLife=100000000,Exp=0.5
TD InitRate=0.050,HalfLife=100000000,Exp=0.5
TD Lambda=0.25 InitRate=0.010,HalfLife=100000000,Exp=0.5
TD Lambda=0.25 InitRate=0.050,HalfLife=100000000,Exp=0.5
TD Lambda=0.5 InitRate=0.010,HalfLife=100000000,Exp=0.5
TD Lambda=0.5 InitRate=0.050,HalfLife=100000000,Exp=0.5
TD Lambda=0.75 InitRate=0.010,HalfLife=100000000,Exp=0.5
TD Lambda=0.75 InitRate=0.050,HalfLife=100000000,Exp=0.5

(d) Done in code: https://github.com/nabilah13/RL-book/blob/master/assignment6/a6p1d.py

From the plot we can see the high variance of Monte Carlo. We can see also that TD(lambda) convergence is in between TD(0) and Monte Carlo, as we expect.