Control HW1 Jieneng Yang

2. Solution:

a. b

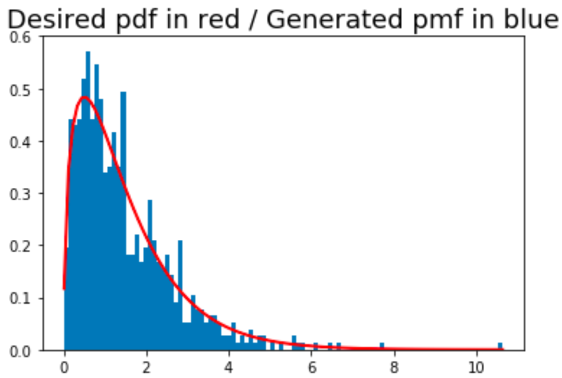
$$f(x) = k\sqrt{x}e^{-x} = x^{\frac{1}{2}}e^{-x} / \Gamma(\tfrac{3}{2}) = \frac{2}{\sqrt{2}} x^{\frac{1}{2}} e^{-x}.$$

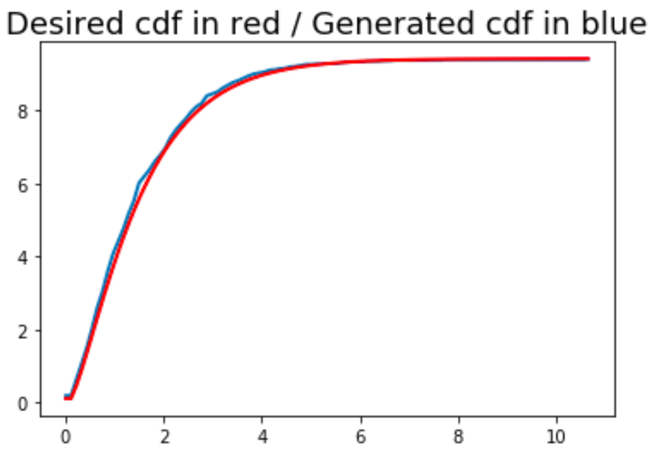
$$g(x) = \lambda e^{-x}.$$

Let $\lambda = \frac{1}{2}$.

$$\frac{f(x)}{g(x)} = \frac{4}{\sqrt{\pi}}\sqrt{x}e^{-\frac{1}{2}x}. \Rightarrow C = 1.36879 \text{ When } x = 1.$$

Python is used to generate the desired gamma variable using acceptance rejection method. (HW_1_2_PY). An empirical histogram of the cdf and comparing with the Gamma cdfs are plotted. We can see two cdfs are almost overlapped, showing that this method is working in the program.





c. $C(x)=\frac{f(x)}{g(x)}=\frac{2\sqrt{x}}{\lambda\sqrt{\pi}}e^{(\lambda-1)x}$.

$$\frac{dc(x)}{dx}=\frac{1}{\lambda\sqrt{\pi}\cdot\sqrt{x}}e^{(\lambda-1)x}+\frac{2\sqrt{x}}{\lambda\sqrt{\pi}}(\lambda-1)e^{(\lambda-1)x}.$$

Let $\frac{dc(x)}{dx}=0 \Rightarrow \frac{1}{\lambda\sqrt{\pi x}}+\frac{2\sqrt{x}}{\lambda\sqrt{\pi}}(\lambda-1)=0$.

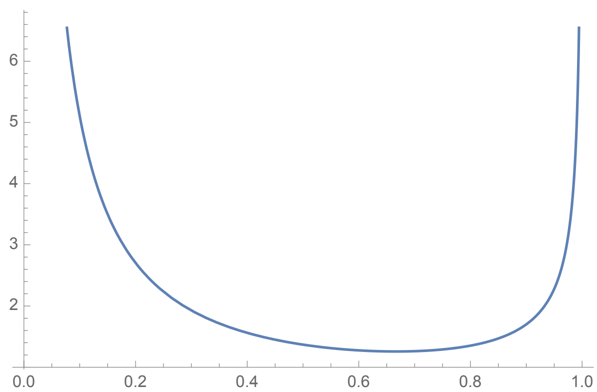
$$\Rightarrow x=\frac{1}{2(1-\lambda)}>0.$$

$$\therefore C(\lambda)=\frac{1}{\lambda}\cdot e^{\frac{-1+\lambda}{2(1-\lambda)}}\cdot\sqrt{\frac{1}{1-\lambda}}\cdot\sqrt{\frac{2}{\pi}}.$$

$$\frac{dc(\lambda)}{\lambda}=0 \rightarrow \lambda=\frac{2}{3}. \quad C\left(\frac{2}{3}\right)=3\sqrt{\frac{3}{2e\pi}}.$$

$\therefore$ When $\lambda=\frac{2}{3}$, the method achieves the minimum expected number of iterations $3\sqrt{\frac{3}{2e\pi}}$.

The plot below shows that the expected iterations reach minimum at x = 2/3



12.

HW1-12. Transition matrix P for regular Markov Chain.

The the rows of U may be taken to be left eigenvectors of P.

$$UPU^{-1} = \Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & \lambda_n \end{pmatrix}$$

We know $\lambda_1 = 1$ as P has an eigenvalue of 1.

$\therefore P = U^{-1}\Lambda U$ Let $U^{-1} = V$.

$$P^t = V\Lambda^t U = V\begin{pmatrix} 1 & & 0 \\ & \lambda_2^t & \\ 0 & & \ddots \; \lambda_n^t \end{pmatrix} U \xrightarrow{t\to\infty} V\begin{pmatrix} 1 & & 0 \\ \vdots & 0 & \vdots \\ & & \ddots \; 0 \end{pmatrix} U = \pi(\infty)$$

Given initial distribution q^0.

generally $q^{(t)} = q^0 P^t = \pi + \sum_{i=2}^{n} \tilde{q}_i \lambda_i^t u_i$.

$(q = \tilde{q}_1 u_1 + \tilde{q}_2 u_2 + \cdots + \tilde{q}_n u_n)$ where $\tilde{q}_i = \frac{q u_i^T}{\|u_i\|^2}$.

implying that $q^{(t)} \xrightarrow[t\to\infty]{} \pi$. then

$\|q^t - \pi\| = O(|\lambda_2|^t)$

This proof works for 2-state Markov chain when n = 2.

In MATLAB, we select

```
P1 = [0.5,0.2,0.3;0.1,0.9,0;0.23,0.23,0.54];
P2 = [0.1,0.6,0.3;0.1,0.9,0;0.1,0.8,0.1];
```

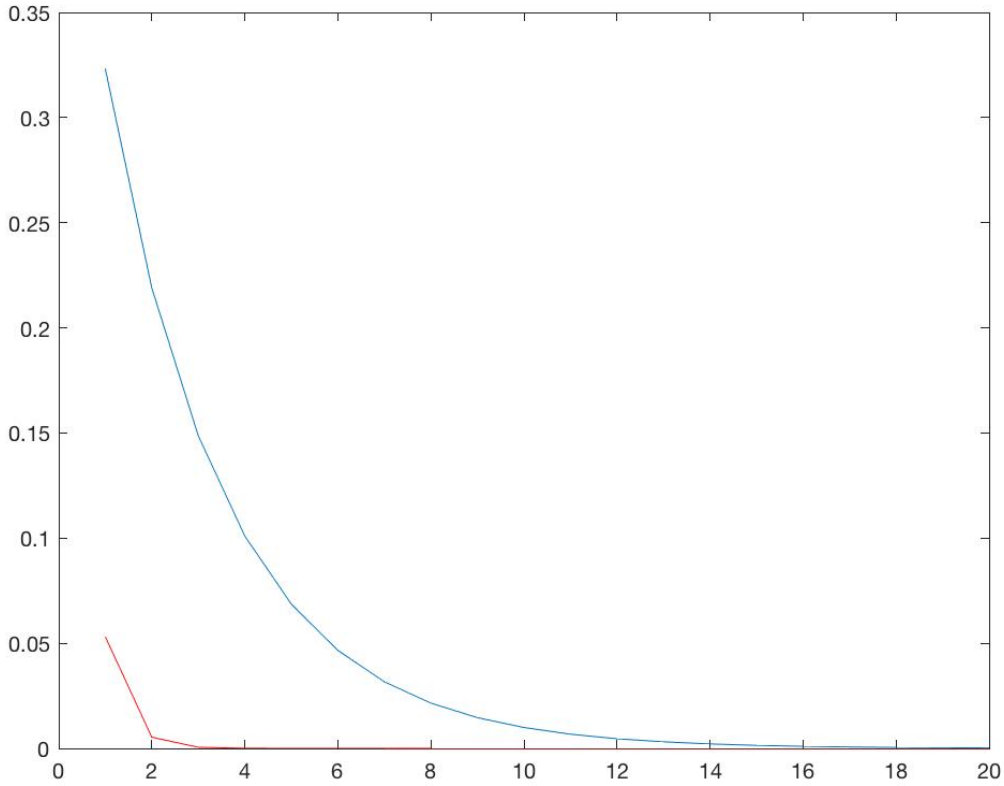
P1 has a bigger second largest value.

```
D1 =

    0.2605         0         0
         0    0.6795         0
         0         0    1.0000
D2 =
```

```
1.0000         0         0
     0    0.0000         0
     0         0    0.1000
```

To compute the converge to the stationary distribution, the L2 norm between each step distribution and the final stationary distribution is plotted, P1 in blue and P2 in red. P2 has faster converge to the stationary distribution, which verifies the conclusion.



13.

Construct A as. HW1-13.

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ \frac{V_1}{V_2}A_{12} & A_{22} & A_{23} & \cdots & A_{2n} \\ \frac{V_1}{V_3}A_{13} & \frac{V_2}{V_3}A_{23} & \cdots & & A_{3n} \\ \vdots & \vdots & & & \vdots \\ \frac{V_1}{V_n}A_{1n} & \frac{V_2}{V_n}A_{2n} & \cdots & & A_{nn} \end{pmatrix}$$

$$V_1' = V_1 A_{11} + V_2 \cdot \frac{V_1}{V_2} A_{12} + \cdots + V_n \cdot \frac{V_1}{V_n} A_{1n}$$

$$= V_1 (A_{11} + A_{12} + \cdots + A_{1n}).$$

$$= V_1 \quad (A_{11} + \cdots + A_{1n} = 1).$$

$$V_2' = V_1 A_{12} + V_2 A_{22} + V_3 \cdot \frac{V_2}{V_3} A_{23} + \cdots + V_n \cdot \frac{V_2}{V_n} A_{2n}.$$

$$= V_2 \cdot \frac{V_1}{V_2} A_{12} + V_2 A_{22} + V_2 \cdot A_{23} + V_2 A_{24} + \cdots + V_2 A_{2n}$$

$$= V_2 \quad (\text{Total(row 2 of } A) = 1)$$

$\cdots$

$\therefore V_m$ can computed as $V_m \times \text{total(row } m \text{ of } A) = V_m$.

$$V_m' = V_1 A_{1m} + V_2 A_{2m} + \cdots + V_{m-1} A_{(m-1)m} + V_m A_{mm}$$

$$+ V_{m+1} \cdot \frac{V_m}{V_{m+1}} A_{m(m+1)} + V_{m+2} \cdot \frac{V_m}{V_{m+2}} A_{m(m+2)} + \cdots + V_n \cdot \frac{V_m}{V_n} A_{mn}$$

$$= V_m \cdot \frac{V_1}{V_m} A_{1m} + V_m \cdot \frac{V_2}{V_m} A_{2m} + \cdots + V_m \frac{V_{m-1}}{V_m} A_{(m-1)m} + V_m \cdot A_{mm}.$$

$$+ V_m A_{m(m+1)} + V_m \cdot A_{m(m+2)} + \cdots V_m \cdot A_{mn}.$$

$$= V_m \cdot (\text{Total (row } m \text{ of } A)) = V_m.$$

$\therefore A\vec{v} = \vec{v}$ $\quad \vec{v}$ stationary distribution.

14.

HW1-14.

No harm to set $i=1$, $j=m$. for m-state set

We need the derive

$$P(x_n=m, x_{n-1}\neq j, \cdots x_1\neq j \mid x_0=1).$$

$\pi(0) = (1, 0, \cdots 0)$

tranistion Matrix $P = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & & & \vdots \\ P_{m1} & \cdots & \cdots & P_{mm} \end{pmatrix}$

Take the first m col and m row. to build another transition Matrix

$$P' = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1(m-1)} \\ P_{21} & & & \vdots \\ \vdots & & & \vdots \\ P_{(m-1)1} & & & P_{(m-1)(m-1)} \end{pmatrix}$$

and $\pi'(0) = (1, 0, \cdots 0)$ $(m-1)$-state set.

then $\pi'(k) = \pi'(0)\cdot P'$ is the state probability when state m is not reachable but the probability is counted.

We can also derive $P_3 = (P_{1m}, P_{2m} \cdots P_{(m-1)m})^T$ represent the probability ~~that fi state~~ of entering state m

$\therefore P(x_n=j, x_{n-1}\neq j \cdots x_1\neq j \mid x_0=i)$

$= \pi'(0)\cdot (P')^{n-1}\cdot P_3$.

16.

The whole process is simulated using Mathematica.

First, construct A as

$$\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Then, only use the first 8 terms in initial distribution pi0 = (1,0,0,0,0,0,0,0,0) and the first 8 rows and 8 cols of A (P). Alos, take the last col of A (fp)

Then the probability that the robot reaches the target at time 10 is

Pi0 * P^9 * fp = 0.0442387

18.

HW1-18. $\frac{Y(z^{-1})}{U(z^{-1})} = \frac{0.5z^{-1} + 0.8z^{-2}}{1 + 0.8z^{-1} + z^{-2}}$.

$A = \begin{bmatrix} -0.8 & -1 \\ 1 & 0 \end{bmatrix}$ $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $c = \begin{pmatrix} 0.5 \\ 0.8 \end{pmatrix}$ $d = 0$.

$$\begin{cases} x(k+1) = A x(k) + b u(k). \\ y(k) = C x(k). \end{cases}$$

covariance $P[0] = C_0$.

$P[k] = AP[k-1]A^T$

$= A^k C_0 A'^k$

$A = \begin{bmatrix} -0.8 & -1 \\ 1 & 0 \end{bmatrix} = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ -0.283 - j0.648 & -0.283 + j0.648 \end{pmatrix}$

$*$ $\begin{pmatrix} -0.4 + j0.9165 & 0 \\ 0 & -0.4 - j0.9165 \end{pmatrix}$

$*$ $\begin{pmatrix} \sqrt{2} + j0.31 & j0.7715 \\ \sqrt{2} - j0.31 & -j0.7715 \end{pmatrix}$

$= H \cdot D \cdot H^{-1}$

D: Diagnomartrix of Eigenvalue of A.

$A^T = (H^{-1})^T \cdot D \cdot H^T$.

$\therefore P[k] = H \cdot D^k \cdot H^{-1} C_0 (H^{-1})^T \cdot D^k H^T$

$P[\infty] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} C_0 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

HW-19.

$m_0 = E[x_0]$.

$m_k = E[x_k]$.

$m_{k+1} = E[a(s_k)]E[x_k]$

$= (a_1 \ a_2)\begin{pmatrix} \pi_1(k) \\ \pi_2(k) \end{pmatrix} m_k$.

or $= (\pi_1(k), \pi_2(k))\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} m_k$.

$(\pi_1(k), \pi_2(k)) = \pi'(k)$

$= \pi'(0) A^k$

$\therefore m_{k+1} = \pi'(0) A^k \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} m_k$

$m_k = (\prod_{i=0}^{k-1} \pi'(0) A^i \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}) \cdot m_0$.

18. Power[A, Infinity] = [0, 1;1, 0]

HW1-20. $y_k = -a_1 y_{k-1} - a_2 y_{k-2} + w_k$

$$\Rightarrow \frac{Y(z^{-1})}{W(z^{-1})} = \frac{z^2}{z^2 + a_1 z + a_2}$$

The transfer function should have poles inside unit circle.

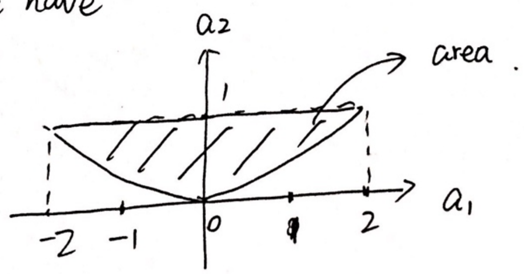
poles: $R \pm jI \Rightarrow a_1 = -2R \quad a_2 = R^2 + I^2$.

$R^2 + I^2 \leq 1$. $\therefore 0 < a_2 < 1$. as $-1 < R < 1$.

$-2 < a_1 < 2$

$\frac{a_1^2}{4} + I^2 < a_2$. $\quad I > 0$.

$\therefore$ We have



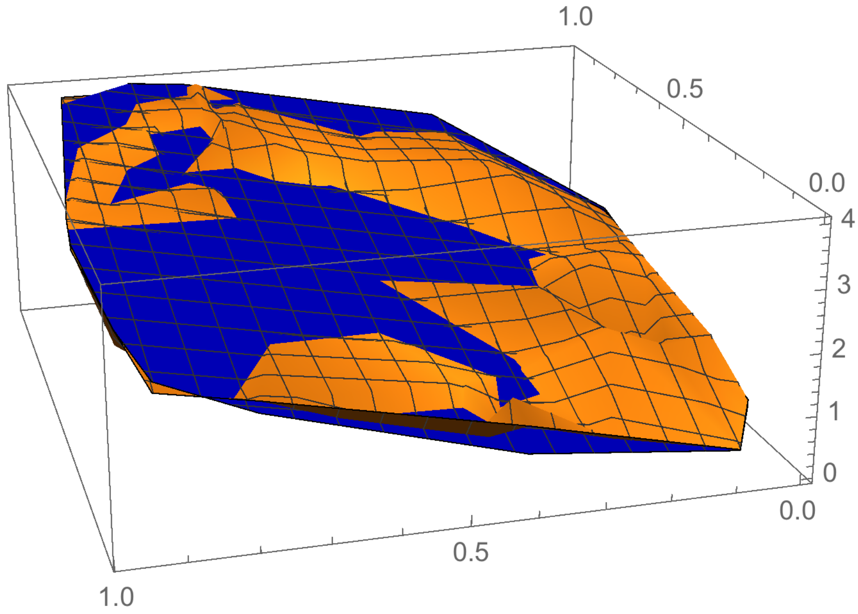
23.

An algorithm is applied in Mathematica. b1 and b2 are generated randomly, and u is a white noise input.

y = b1*u(k-1) + b2*u2(k-2) + white noise

Then FindMinimum[] function is used to find the local minimum value of the absolute error, which can find the least absolute estimate.

Then the 3D plots of generated data in orange and estimate data in blue are shown below. The estimate plane fits the original data plane, which verifies the method.


1.0
0.5
0.0
4
3
2
1
0
0.0
0.5
1.0