Homework # 1

Pittsburgh Campus
Jingyi Guo

1.

a) $\int_0^x k\lambda(\lambda t)^{k-1}e^{-(\lambda t)^k}dt$ let $y=\lambda t$

$= \int_0^{\lambda x} k y^{k-1} e^{-y^k} dy$

$= -e^{-y^k}\Big|_0^{\lambda x}$ $\therefore F_X(x) = \begin{cases} 1-e^{-(\lambda x)^k} & , x \geq 0 \\ 0 & , x<0 \end{cases}$

$= 1-e^{-(\lambda x)^k}$

Solve $u = F(x) = 1-e^{-(\lambda x)^k}$

$\therefore 1-u = e^{-(\lambda x)^k}$

$\therefore -(\lambda x)^k = \ln(1-u)$

$\therefore x = \frac{1}{\lambda}(-\ln(1-u))^{\frac{1}{k}} = F^{-1}(u)$

If $U \sim U(0,1)$, we define $X = \frac{1}{\lambda}[-\ln(1-u)]^{\frac{1}{k}}$ to generate this dist.

b) $X = F^{-1}(u) = \inf\{x \mid u \leq 1-e^{-2x(x-b)/h}\}$ Since $\lim_{x\to\infty} F_X(x) = \lim_{x\to\infty} 1-e^{-2x(x-b)/h} = 1$

we have : $h>0$.

Consider $u \leq 1-e^{-2x(x-b)/h}$

$e^{-2x(x-b)/h} \leq 1-u$

$\frac{-2x(x-b)}{h} \leq \ln(1-u)$

$\therefore x^2 - bx + \frac{h\ln(1-u)}{2} \geq 0$

$\therefore x \leq \frac{b-\sqrt{b^2-2h\ln(1-u)}}{2}$ or $x \geq \frac{b+\sqrt{b^2-2h\ln(1-u)}}{2}$

Since $\frac{b-\sqrt{b^2-2h\ln(1-u)}}{2} \leq \frac{b}{2} < b \leq \max(0,b)$, we have $X = F^{-1}(u) = \frac{b+\sqrt{b^2-2h\ln(1-u)}}{2}$

Since $u \in (0,1)$, $\ln(1-u)<0$, $\sqrt{b^2-2h\ln(1-u)} > |b|$, $X > \frac{b+|b|}{2} = \max(b,0)$

$\therefore$ If $U \sim U(0,1)$, we define $X = \frac{b+\sqrt{b^2-2h\ln(1-u)}}{2}$

c) $\int_{-\infty}^{x} \frac{1}{\pi}\frac{1}{1+t^2}dt = \frac{1}{\pi}\arctan t\Big|_{-\infty}^{x} = \frac{1}{\pi}\arctan x + \frac{1}{2}$

Solve $u = F(x) = \frac{1}{\pi}\arctan x + \frac{1}{2}$

$\therefore \pi u = \arctan x + \frac{\pi}{2}$

$\therefore x = \tan(\pi u - \frac{\pi}{2}) = -\cot(\pi u)$

If $U \sim U(0,1)$, we define $X = -\cot(\pi u)$ to generate this dist.

d) i) Solve $u = F(x) = \exp(-\exp(-x))$

$\ln u = -\exp(-x)$

$\ln(-\ln u) = -x$

$\therefore x = -\ln(-\ln u) = F^{-1}(u)$

$\therefore$ If $U \sim U(0,1)$, we define $X = -\ln(-\ln(u))$ to generate this dist.

ii) $P(M_n - a_n \leq x) = P(M_n \leq x+a_n) = P((X_1 \leq x+a_n)\cap\cdots\cap(X_n \leq x+a_n)) = P(X_1 \leq x+a_n)\cdots P(X_n \leq x+a_n))$

$= \exp(-\exp(-x-a_n))\cdots\exp(-\exp(-x-a_n))$

$= \exp(-n\exp(-x-a_n))$

Let $\exp(-n\exp(-x-a_n)) = \exp(-\exp(-x))$; we have $-n\exp(-x-a_n) = -\exp(-x)$

(see back)

$\therefore \ln n - x - a_n = -x$

$\therefore a_n = \ln n$

$\therefore M_n - \ln n$ has a standard Gumbel dist.

So M_n is increasing in n, but the growth rate is decreasing (since $\frac{d}{dn}(\ln n) = \frac{1}{n}$ is decreasing in n)

Simulation Methods for Option Pricing

Jingyi Guo (jingyig1)
Pittsburgh Campus

2.

a)

Simulation result:

```
transmat =

    0.5692    0.3005    0.0845    0.0241    0.0126    0.0060    0.0006    0.0025
    0.0275    0.6190    0.2407    0.0670    0.0211    0.0163    0.0017    0.0067
    0.0053    0.0954    0.6088    0.1848    0.0523    0.0317    0.0037    0.0180
    0.0030    0.0295    0.1907    0.4873    0.1411    0.0815    0.0116    0.0553
    0.0017    0.0131    0.0562    0.1629    0.3386    0.2205    0.0333    0.1736
    0.0004    0.0078    0.0214    0.0432    0.1074    0.4294    0.0590    0.3315
    0.0003    0.0041    0.0295    0.0345    0.0469    0.1210    0.1279    0.6358
         0         0         0         0         0         0         0    1.0000
```

Compute the standard errors for transmat in a):

```
stderr =

   1.0e-03 *

    0.4952    0.4585    0.2781    0.1533    0.1114    0.0772    0.0254    0.0499
    0.1635    0.4856    0.4275    0.2500    0.1437    0.1268    0.0407    0.0816
    0.0724    0.2938    0.4880    0.3882    0.2225    0.1752    0.0609    0.1331
    0.0545    0.1691    0.3929    0.4998    0.3482    0.2736    0.1073    0.2285
    0.0417    0.1137    0.2304    0.3693    0.4732    0.4146    0.1795    0.3787
    0.0196    0.0881    0.1446    0.2032    0.3096    0.4950    0.2355    0.4707
    0.0170    0.0639    0.1692    0.1826    0.2115    0.3261    0.3340    0.4812
         0         0         0         0         0         0         0         0
```

Compare to exp(Q*T):

```
P5_2 =

    0.5685    0.3007    0.0845    0.0242    0.0125    0.0062    0.0006    0.0025
    0.0274    0.6197    0.2403    0.0669    0.0210    0.0164    0.0016    0.0067
    0.0054    0.0954    0.6090    0.1846    0.0523    0.0317    0.0037    0.0181
    0.0030    0.0295    0.1907    0.4862    0.1418    0.0814    0.0116    0.0557
    0.0017    0.0131    0.0564    0.1627    0.3385    0.2204    0.0333    0.1738
    0.0004    0.0079    0.0216    0.0432    0.1074    0.4301    0.0591    0.3303
    0.0003    0.0041    0.0297    0.0346    0.0468    0.1209    0.1277    0.6359
         0         0         0         0         0         0         0    1.0000
```

My simulation result is pretty similar to the arithmetic result. And the standard error of my simulation is quite small.

b)

Simulation result:

```
transmat =

    0.6815    0.2579    0.0403    0.0096    0.0091    0.0013    0.0001    0.0001
    0.0226    0.7261    0.1962    0.0356    0.0102    0.0087    0.0002    0.0006
    0.0032    0.0790    0.6862    0.1761    0.0348    0.0156    0.0007    0.0043
    0.0024    0.0218    0.1781    0.5239    0.1784    0.0609    0.0095    0.0249
    0.0017    0.0117    0.0501    0.1673    0.3407    0.2351    0.0526    0.1409
    0.0002    0.0061    0.0134    0.0309    0.1361    0.4500    0.1127    0.2506
    0.0006    0.0070    0.0254    0.0369    0.0553    0.0894    0.0947    0.6907
         0         0         0         0         0         0         0    1.0000
```

Compute the standard errors for transmat in b):

```
stderr =

   1.0e-03 *

    0.4659    0.4375    0.1967    0.0977    0.0952    0.0358    0.0079    0.0120
    0.1486    0.4460    0.3971    0.1852    0.1003    0.0928    0.0144    0.0237
    0.0564    0.2698    0.4640    0.3809    0.1834    0.1239    0.0273    0.0654
    0.0486    0.1462    0.3826    0.4994    0.3829    0.2392    0.0972    0.1558
    0.0407    0.1076    0.2182    0.3732    0.4740    0.4240    0.2232    0.3479
    0.0154    0.0781    0.1148    0.1729    0.3429    0.4975    0.3163    0.4334
    0.0253    0.0836    0.1572    0.1885    0.2285    0.2853    0.2927    0.4622
         0         0         0         0         0         0         0         0
```

Compare with result in a):

The transition probabilities matrix for b) do not differ very much from the matrix for a). Note that the standard errors of the transition probabilities matrix for b) are generally smaller than that for a).

Code for 2:

```
Q = [-.1154, .1019, .0083, .0020, .0031, 0, 0, 0;
    .0091, -.1043, .0787, .0105, .0030, .0030, 0, 0;
    .0010, .0309, -.1172, .0688, .0107, .0048, 0, .0010;
    .0007, .0047, .0713, -.1711, .0701, .0174, .0020, .0049;
    .0005, .0025, .0089, .0813, -.2530, .1181, .0144, .0273;
    0, .0021, .0034, .0073, .0568, -.1928, .0479, .0753;
    0,0,.0142, .0142, .0250, .0928, -.4318, .2856;
    0, 0, 0, 0, 0, 0, 0, 0];
M = [0,.8838, .0720, .0173, .0269, 0, 0, 0;
    .0872, 0, .7545, .1007, .0288, .0288, 0,0;
    .0085, .2637, 0, .5870, .0913, .0410, 0, .0085;
    .0041, .0275, .4167, 0, .4097, .1017, .0117, .0286;
    .0020, .0099, .0352, .3213, 0, .4668, .0569, .1079;
    0, .0109, .0176, .0379, .2946, 0, .2484, .3906;
    0, 0, .0329, .0329, .0579, .2149, 0, .6614;
```

```
    0,0,0,0,0,0,0,1];
lam = [.1154, .1043, .1172, .1711, .2530, .1929, .4318, .0001];

% (a)
% S1. create aliasing table
AliasTable=zeros(56,3);
% aliasing table for state k : row 1+(k-1)*7 ~ 7+(k-1)*7
for k=1:8
    one2seven=[1;2;3;4;5;6;7;8];
    L=[one2seven,transpose(M(k,:))];
    N=size(L,1);
    L(:,2)=(N-1)*L(:,2);
    T=zeros(N-1,3);
    for i =1:N-1
        L=sortrows(L(1:N-i+1,:),2);
        T(i,1)=L(1,2);
        T(i,2)=L(1,1);
        T(i,3)=L(N-i+1,1);
        L(N-i+1,2)=L(N-i+1,2)-1+L(1,2);
        for j=2:N-i+1
            L(j-1,1)=L(j,1);
            L(j-1,2)=L(j,2);
        end
    end
    AliasTable((1+(k-1)*7):(7+(k-1)*7),:)=T;
end

% S2. loop through
n=1000000; % generate 1000,000 paths

for k=1:8 % states
    for i=1:n
        % before time expires
        % clock=(-1/lam(k)*log(rand));
        % (b)
        clock=1/2*((-1/lam(k)*log(rand))+(-1/lam(k)*log(rand)));
        state=k;
        while clock<5
            % random select next state
            T=AliasTable((1+(state-1)*7):(7+(state-1)*7),:);
            P=transpose(T(:,1));
            X=transpose(T(:,2));
```

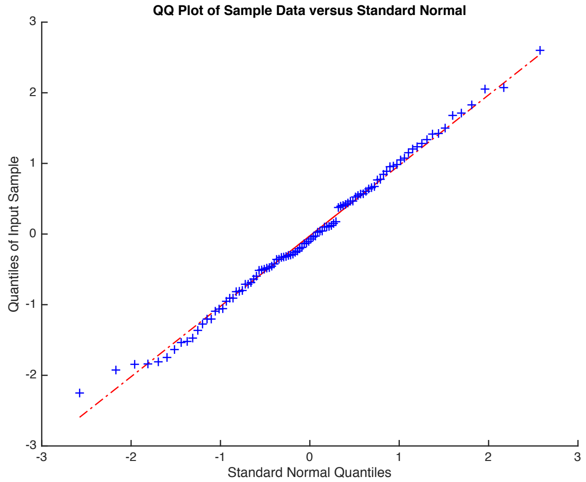
```
            A=transpose(T(:,3));
            V=(8-1)*rand(1);
            I=ceil(V);
            W=I-V;
            Y=(W<=P(I));
            X=X(I)*Y+A(I)*(1-Y);
            state=X;
            %clock=clock+(-1/lam(state)*log(rand));  % holding time
            % (b)
            clock=clock+1/2*((-1/lam(k)*log(rand))+(-
1/lam(k)*log(rand)));
        end
        count(state,k)=count(state,k)+1;
    end
end

transmat=transpose(count)/n;
%std. error matrix
stderr=sqrt(n*transmat.*(1-transmat))/n;


%P5_2=expm(Q*5)
```

3.

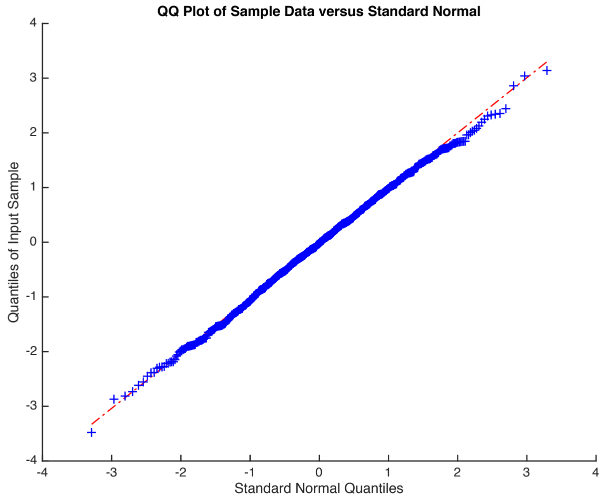
a) code:

```
% (a) normal generator
X100 = normrnd(0,1,[1,100]);
X1000 = normrnd(0,1,[1,1000]);
X10000 = normrnd(0,1,[1,10000]);
qqplot(X100);
%qqplot(X1000);
%qqplot(X10000);
```

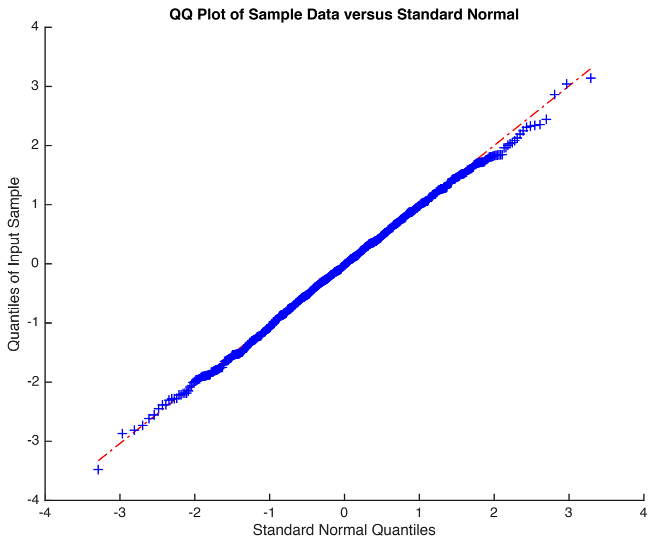
Q-Q plot of X100:



Q-Q plot of X1000:



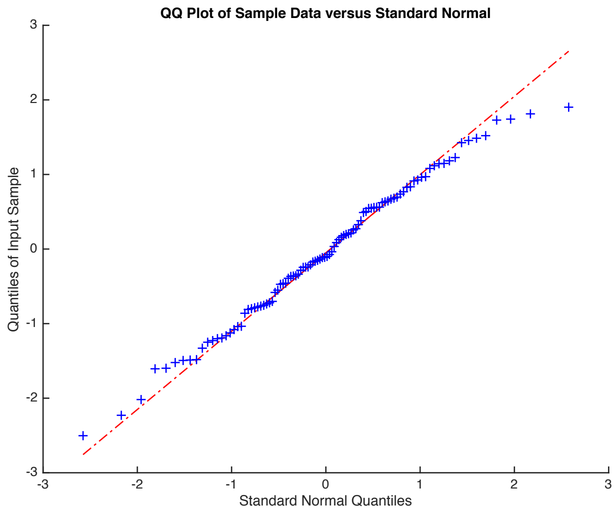
Q-Q plot of X1000:



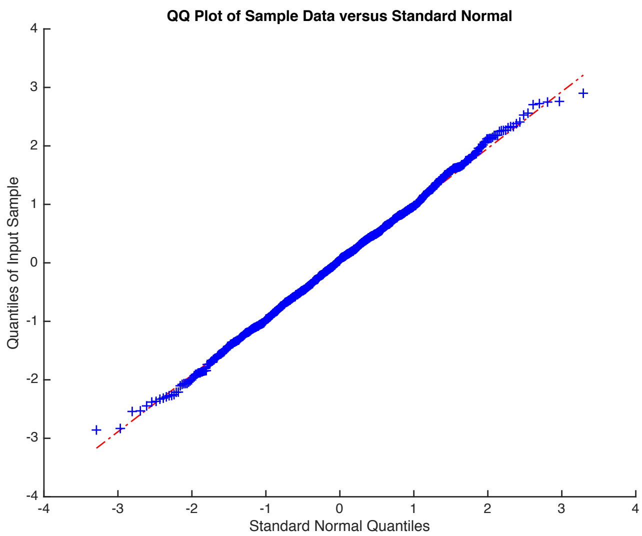
b) code:

```
Y100=zeros(1,100);
for i = 1:100
    sumu=0;
    for j=1:12
        sumu=sumu+rand;
    end
    Y100(i)=sumu-6;
end

Y1000=zeros(1,1000);
for i = 1:1000
    sumu=0;
    for j=1:12
        sumu=sumu+rand;
    end
    Y1000(i)=sumu-6;
end

Y10000=zeros(1,10000);
for i = 1:10000
    sumu=0;
    for j=1:12
        sumu=sumu+rand;
    end
    Y10000(i)=sumu-6;
end
qqplot(Y100);
%qqplot(Y1000);
%qqplot(Y10000);
```

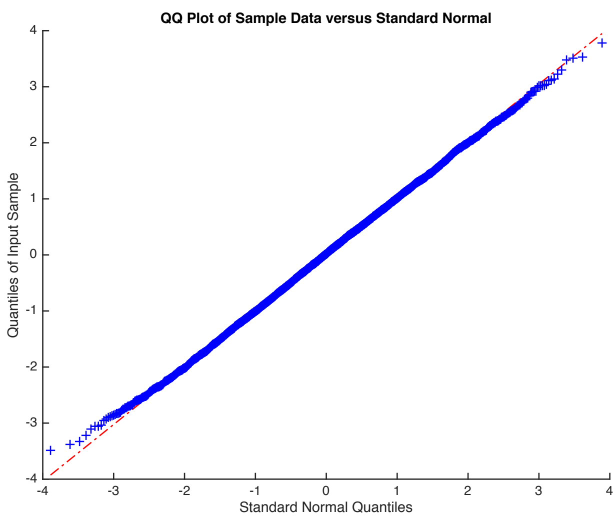
Q-Q plot of Y100:



Q-Q plot of Y1000:



Q-Q plot of Y10000:



Comment on the quality of methods:
Based on Q-Q plots, we can see that: for 100 observations, the “poor man’s” generator does not work very well since the many points (esp. the tails) do not align with the 45 degree line. But for 1000 and 10000 observations, the performance of “poor man’s” generator is acceptable.