Question 5

(a)

$$(a)\quad -\log J \sim Exp(u_d)$$

$$P[-\log J < x] = \begin{cases} \frac{1}{u_d} e^{-\frac{x}{u_d}} & x \in [0, +\infty) \\ 0. & o.w. \end{cases}$$

$$P[\log J \le x]$$
$$P[-\log J \ge -x]$$
$$= 1 - P[-\log J \le -x] = e^{\frac{1}{u_d} x}.$$

$$P_{\log(J)}(x) = \begin{cases} \frac{1}{u_d} e^{\frac{1}{u_d} x} & x \in [-\infty, 0) \\ 0 & o.w. \end{cases}$$

$$E[J_u] = E[e^x] = \int_0^{+\infty} e^x \cdot \frac{1}{u_n} e^{-\frac{x}{u_n}} = \int_0^{+\infty} \frac{1}{u_n} exp(x \cdot \frac{u_n - 1}{u_n}) dx$$
$$= \frac{1}{u_n - 1} \int_0^{+\infty} \frac{u_n - 1}{u_n} exp(x \cdot \frac{u_n - 1}{u_n}) dx$$
$$= \frac{1}{u_n - 1}(0 - 1) = \frac{1}{1 - u_n}.$$

$$E[J_d] = [e^x] = \int_{-\infty}^{0} e^x \cdot \frac{1}{u_d} e^{\frac{x}{u_d}} = \int_{-\infty}^{0} \frac{1}{u_d} exp(x \cdot \frac{1 + u_d}{u_d}) dx$$
$$= \frac{1}{1 + u_d}$$

$$K = E[J - 1] = E[J] - 1$$
$$= P_u E[J_u] + P_d E[J_d] - 1.$$
$$= \frac{P_u}{u_n - 1} + \frac{P_d.}{u_d + 1} - 1.$$

(b)

```
T = 1.00;                   % time to expiry
sigma = 0.15;               % volatility
r = 0.03;                   % risk free interest rate
S_init = 90;
K = S_init;
%jump size: log normal distribution
p_u = 0.4;                  % probability of up jump
lambda = .1;                  % jump size arrival rate lambda = 0.1
```

```
u_u = 0.32;             % parameter for up movement
u_d = 0.3;              % parameter for down movement
N_sim = 25000;
N=800;
delt = T/N;         % delt = 1/1000

jump_up_mean = u_u;
jump_down_mean = u_d;
%
% compenstaed drift E[J-1]
%

%kappa value that we found int question 5 (a)
kappa = p_u/(1-u_u) + (1-p_u)/(u_d + 1) - 1;
%
% compensated drift for X = log(S)

drift = r - delt/2 - lambda*kappa;

%
% X = log(S)
%

X_old(1:N_sim,1) = log(S_init);
X_new(1:N_sim,1) = zeros(N_sim,1);

jump_chek = zeros(N_sim,1);
jump_size = zeros(N_sim,1);
jump_mask = zeros(N_sim,1);

for i = 1:N %timestep loop


    jump_chek(:,1) = rand(N_sim,1);                         % first check to
determine lambda*delt
    jump_chek2(:, 1) = rand(N_sim, 1);                      % second check to
determine p_u
    jump_mask(:,1) = (jump_chek(:,1) <= lambda*delt);
    jump_mask2(:, 1) = (jump_chek2(:, 1) <= p_u);
    jump_mask3(:, 1) = (jump_chek2(:, 1) > p_u);
    jump_size = (jump_mask2 .* exprnd(u_u, N_sim, 1)) -...  % determine the
jump_size
        (jump_mask3 .* exprnd(u_d, N_sim, 1));
    jump_size = jump_size .* jump_mask;

    X_new(:,1) = X_old(:,1) + drift*delt + sigma*sqrt(delt)*randn(N_sim,1)+...
        jump_size(:,1);
    X_old(:,1) = X_new(:,1);
end %timestpe loop

S(:,1) = exp(X_new(:,1));

n_bin = 200;
hist(S,n_bin);

W(:,1) = max(K-S,0);
%European put option
Ep = exp(-r*T)*mean(W(:,1));
disp(sprintf('put option price is : %.5g\n', Ep));
```

	800/25000	1600/100000	3200/400000
European Put	5.0515	5.0633	5.0722

Table: European Put Option price under different timesteps and simulation number

(c)

```
T = 1.00;                  % time to expiry
sigma = 0.15;              % volatility
r = 0.03;                  % risk free interest rate
mu = 0.10;
S_init = 90;
K=linspace(70, 120, 20);
%jump size: log normal distribution
p_u = 0.4;                % probability of up jump
lambda = .1;                % jump size arrival rate lambda = 0.1



u_u = 0.32;                % parameter for up movement
u_d = 0.3;                 % parameter for down movement
N_sim = 25000;
N=800;
delt = T/N;            % delt = 1/1000

jump_up_mean = u_u;
jump_down_mean = u_d;
jump_up_vol = u_u^2;
jump_down_vol = u_d^2;
%
% compenstaed drift E[J-1]
%

%kappa = exp(.5*jump_vol*jump_vol + jump_mean) - 1.;
kappa = p_u/(1-u_u) + (1-p_u)/(u_d + 1) - 1;
%
% compensated drift for X = log(S)

drift = r - delt/2 - lambda*kappa;
X_old = zeros(N_sim,20);
X_new = zeros(N_sim,20);
S = zeros(N_sim,20);
W = zeros(N_sim,20);
P = zeros(1,20);
%
% X = log(S)
%
for j = 1:20
X_old(1:N_sim,j) = log(S_init);
X_new(1:N_sim,j) = zeros(N_sim,1);

jump_chek = zeros(N_sim,1);
jump_size = zeros(N_sim,1);
jump_mask = zeros(N_sim,1);

for i = 1:N %timestep loop
```

```
    jump_chek(:,1) = rand(N_sim,1);
    jump_chek2(:, 1) = rand(N_sim, 1);
    jump_mask(:,1) = (jump_chek(:,1) <= lambda*delt);
    jump_mask2(:, 1) = (jump_chek2(:, 1) <= p_u);
    jump_mask3(:, 1) = (jump_chek2(:, 1) > p_u);
    jump_size = (jump_mask2 .* exprnd(u_u, N_sim, 1)) -...
        (jump_mask3 .* exprnd(u_d, N_sim, 1));
    jump_size = jump_size .* jump_mask;


    jump_size = jump_size.*jump_mask;

    X_new(:,j) = X_old(:,j) + drift*delt + sigma*sqrt(delt)*randn(N_sim,1)+...
        jump_size(:,1);
    X_old(:,j) = X_new(:,j);
end %timestpe loop

S(:,j) = exp(X_new(:,j));

n_bin = 200;
hist(S,n_bin);

W(:,j) = max(K(j)-S(:,j),0);
%European put option
P(j) = mean(W(:,j));
disp(sprintf('put option price is : %.5g\n', P(j)));
end
Volat = blsimpv(S_init,linspace(70,120,20),r,T,P,[],[],[],false);
plot(linspace(70,120,20),Volat)
xlabel('Strike Price');
ylabel('Implied Volatility');
```

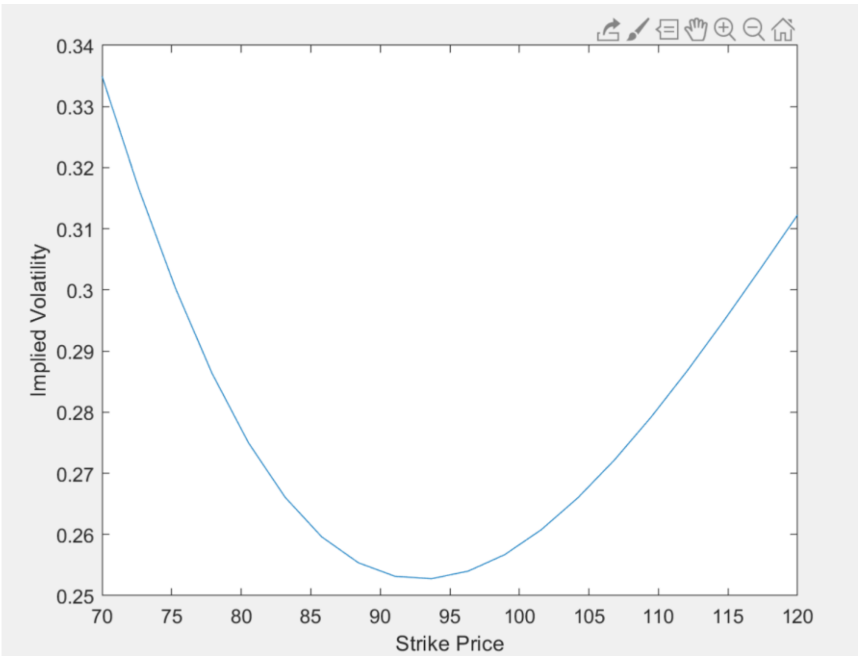

Figure. The Implied Volatility vs the Strike Price

It has shown in the graph, the observed volatility forms a volatility smile over the European Option price. The graph generated by the simulation is reasonable to the implied volatility decreases first and then increases with the increase of strike prices