Option Pricing Using Bionomial Lattice Method

Parameter for options and underlying asset

```
r = 0.02;       %risk-free rate
S0 = 100;       %current price of the underlying stock
sig = 0.25;     %volatility of stock
K = 105;        %strike price
steps = 8;      %number of step to compute (2 months x 4 week/month)
T = 8/52;       %time to maturity in years
dt = T/steps;   %simulation unit time in years
```

```
%return
R = exp(r*dt);
%up factor
u = exp(sig*sqrt(dt))
```

```
u = 1.0353
```

```
%down factor
d = 1/u
```

```
d = 0.9659
```

```
%risk neutral probability
q = (R-d)/(u-d)
```

```
q = 0.4969
```

```
%crete the matrix to store price lattice
priceLattice = nan(steps+1,steps+1);
priceLattice(1,1) = S0;

% Loop over each node to calculate underlying price lattice
for idx = 2:steps+1
    priceLattice(1:idx-1,idx) = priceLattice(1:idx-1,idx-1)*u;
    priceLattice(idx,idx) = priceLattice(idx-1,idx-1)*d;
end

priceLattice
```

```
priceLattice = 9×9
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  123.1229  127.4662 ···
       NaN   96.5925  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275
       NaN       NaN   93.3012   96.5925  100.0000  103.5277  107.1798  110.9607
       NaN       NaN       NaN   90.1220   93.3012   96.5925  100.0000  103.5277
       NaN       NaN       NaN       NaN   87.0511   90.1220   93.3012   96.5925
       NaN       NaN       NaN       NaN       NaN   84.0848   87.0511   90.1220
       NaN       NaN       NaN       NaN       NaN       NaN   81.2197   84.0848
       NaN       NaN       NaN       NaN       NaN       NaN       NaN   78.4521
       NaN       NaN       NaN       NaN       NaN       NaN       NaN       NaN
```

```
% Calculate the value at expiry
```

```
valueLattice = nan(size(priceLattice));
valueLattice(:,end) = max(K-priceLattice(:,end),0);
```

```
% backward pass to get values at the earlier times for american option
steps = size(priceLattice,2)-1;
for idx = steps:-1:1

    %calculate the option value
    valueLattice(1:idx,idx) = exp(-r*dt)*(q*valueLattice(1:idx,idx+1) + (1-q)*valueLattice(2:id

    %if payoff from exercising is greater than value of option then
    %exercise and set the value of the option as payoff
    valueLattice(1:idx,idx) = max(K-priceLattice(1:idx,idx), valueLattice(1:idx,idx));
end

valueLattice
```

```
valueLattice = 9×9
    6.8868    4.5497    2.6062    1.1734    0.3199         0         0         0 ···
       NaN    9.2003    6.4725    4.0233    2.0172    0.6360         0         0
       NaN       NaN   11.9013    8.8963    6.0075    3.3828    1.2647         0
       NaN       NaN       NaN   14.8780   11.7562    8.6042    5.4773    2.5146
       NaN       NaN       NaN       NaN   17.9489   14.8780   11.6988    8.4075
       NaN       NaN       NaN       NaN       NaN   20.9152   17.9489   14.8780
       NaN       NaN       NaN       NaN       NaN       NaN   23.7803   20.9152
       NaN       NaN       NaN       NaN       NaN       NaN       NaN   26.5479
       NaN       NaN       NaN       NaN       NaN       NaN       NaN       NaN
```

```
American_Put_price = valueLattice(1)
```

```
American_Put_price = 6.8868
```

```
% extract paths from bionomial lattice
% helper function walk is used to iterate from node to node to obtain paths
global pricePaths;
pricePaths = [];
start_node = [1 , 1];
walk(start_node,[], priceLattice, steps);
pricePaths
```

```
pricePaths = 256×9
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  123.1229  127.4662 ···
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  123.1229  127.4662
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  123.1229  118.9275
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  123.1229  118.9275
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  114.8751  118.9275
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  114.8751  118.9275
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  114.8751  110.9607
  100.0000  103.5277  107.1798  110.9607  114.8751  118.9275  114.8751  110.9607
  100.0000  103.5277  107.1798  110.9607  114.8751  110.9607  114.8751  118.9275
  100.0000  103.5277  107.1798  110.9607  114.8751  110.9607  114.8751  118.9275
     ⋮
     ⋮
```

```
% calculate the probabailty for each end node
% there are multiple paths leading to the same end node.
% all the paths leading to same nodes have same probabiltiy
for i = 1:steps+1
    prob(i) = q^(steps+1-i)*((1-q)^(i-1));
end
prob
```

```
prob = 1×9
    0.0037    0.0038    0.0038    0.0039    0.0039    0.0040    0.0040    0.0041 ···
```

```
n_paths = 2^steps;
p_index = nan(n_paths,1);

% create probability vector to multiply with payoffs to find the expected
% payoff.
for j = 1:steps+1
    I = find(pricePaths(:,end) == priceLattice(j,end));
    p_index(I) = prob(j);
end
p_index
```

```
p_index = 256×1
    0.0037
    0.0038
    0.0038
    0.0038
    0.0038
    0.0038
    0.0038
    0.0039
    0.0038
    0.0038
      ⋮
      ⋮
```

Asian Call price

```
Asian_call_payoffs = (mean(pricePaths, 2) - K).*((mean(pricePaths, 2) - K) > 0);
Asian_call_price = sum(Asian_call_payoffs.*p_index)*exp(-r*T)
```

```
Asian_call_price = 0.6251
```

Asian Put price

```
Asian_put_payoffs = ( K - mean(pricePaths, 2)).*((K - mean(pricePaths, 2)) > 0);
Asian_put_price = sum(Asian_put_payoffs.*p_index)*exp(-r*T)
```

```
Asian_put_price = 5.4562
```

Lookback Call price

```
lb_call_payoffs = (max(pricePaths,[], 2) - K).*((max(pricePaths,[], 2) - K) > 0);
lb_call_price = sum(lb_call_payoffs.*p_index)*exp(-r*T)
```

```
lb_call_price = 3.3246
```

Lookback Put price

```
lb_put_payoffs = (K - min(pricePaths,[], 2)).*((K - min(pricePaths,[], 2)) > 0);
lb_put_price = sum(lb_put_payoffs.*p_index)*exp(-r*T)
```

```
lb_put_price = 11.0163
```

Floating Lookback Call price

```
flb_call_payoffs = (pricePaths(:,end) - min(pricePaths,[], 2)).*((pricePaths(:,end) - min(price
flb_call_price = sum(flb_call_payoffs.*p_index)*exp(-r*T)
```

```
flb_call_price = 6.3388
```

Floating Lookback Put price

```
flb_put_payoffs = (max(pricePaths,[], 2) - pricePaths(:,end)).*((max(pricePaths,[], 2) - priceP
flb_put_price = exp(-r*T)*mean(flb_put_payoffs)
```

```
flb_put_price = 6.2133
```