

金融商品設計與評價

匯率相關金融商品 I

壹、 外匯交換契約之意義與功能

外匯交換契約的出現比利率交換契約早，其成長落後於後者，但目前市場規模兩者相當。兩者的概念很相似，後者指的是在契約到期前，依照契約所訂定的時段支付或收取按照名目本金計算所得到的差額。在利率交換契約中會有交換率的存在，其通常指的是固定利率，而交換率在利率債券為平價發行時會相等。利率交換契約像是把多個遠期契約組合在一起，因此其在簽的當下，價值為零，但只要契約一完成便有其價值，並可透過依照浮動利率及固定利率的折現因子評價過去所發行的利率交換。而外匯交換契約與利率交換最大的差異便是交換的標的，前者是以本國匯率與他國匯率作為標的而後者是以固定利率與浮動利率。

(一) 平行貸款

為外幣交換的前身，當美國的A公司與法國的B公司簽訂契約，A公司將美元資金提供B公司在美國的子公司，同時B公司亦將等值的法郎資金提供A公司在法國的子公司，便為「平行貸款」之由來。但這樣的安排有缺點存在像是借出去的錢必須出現在母公司的資產負債表，而當其中一方違約時，另一方無法免除債務。

(二) 外匯交換契約

因為有上文所提到的缺點存在，因而生成，透過銀行作為仲介，避免信用風險，並因為不是借貸行為，不用寫在資產負債表上，但目前必須附註揭露。

(三) 外幣交換契約的應用

除了突破了匯管制，並整合國際金融市場，可以在某些幣別取得較低廉的資金，再透過外幣交換契約換成所需貨幣、賺取匯兌利差、規避外匯風險。

貳、 問題

(1) 為甚麼外匯交換契約出現的明明比利率交換早，其成長卻落後？

```
%LatticeEurCall.m
function [price,lattice] =
LatticeEurCall(S0,X,r,T,sigma,N)
    deltaT = T/N;
    u = exp(sigma*sqrt(deltaT));
    d = 1/u; % ud = 1;
    p = (exp((r)*deltaT) - d)/(u-d);
    %p = (exp(rd-rf)*dt-d)/(u-d);
    %p = (1-d)/(u-d);
    lattice = zeros(N+1, N+1);
    for j = 0:N
        lattice(N+1,j+1) = max(0,S0*(u^j)*(d^(N-j))-X);
    end
    for i = N-1:-1:0
        for j = 0:i
            lattice(i+1,j+1) = exp(-
r*deltaT)*(p*lattice(i+2,j+2)+(1-p)*lattice(i+2,j+1));
        end
    end
    price = lattice(1,1)
end
%LatticeEurPut
function [price,lattice] =
LatticeEurPut(S0,X,r,T,sigma,N)
    deltaT = T/N;
    u = exp(sigma*sqrt(deltaT));
    d = 1/u; % ud = 1;
    p = (exp(r*deltaT) - d)/(u-d);
    lattice = zeros(N+1, N+1);
    for j = 0:N
        lattice(N+1,j+1) = max(0,X-S0*(u^j)*(d^(N-j)));
    end
    for i = N-1:-1:0
        for j = 0:i
            lattice(i+1,j+1) = exp(-
r*deltaT)*(p*lattice(i+2,j+2)+(1-p)*lattice(i+2,j+1));
        end
    end
    price = lattice(1,1)
end
```

```

%LatticeAmCall
function [price,lattice] =
LatticeAmCall(S0,X,r,T,sigma,N)
deltaT = T/N;
u = exp(sigma*sqrt(deltaT));
d = 1/u;
p = (exp(r*deltaT) - d)/(u-d);
%p = (exp(rd-rf)*dt-d)/(u-d);
%p = (1-d)/(u-d);
lattice = zeros(N+1,N+1);
for j = 0:N
    lattice(N+1,j+1) = max(0,S0*(u^j)*(d^(N-j))-X);
end
for i = N-1:-1:0
    for j = 0:i
        lattice(i+1,j+1) = max(S0*(u^j)*(d^(i-j))-
X,exp(-r*deltaT)*(p*lattice(i+2,j+2)+(1-
p)*lattice(i+2,j+1)));
    end
end
price = lattice(1,1);
end

%LatticeAmPut
function [price,lattice] =
LatticeAmPut(S0,X,r,rf,T,sigma,N,last)
deltaT = T/N;
u = exp(sigma*sqrt(deltaT));
d = 1/u;
p = (exp((r-rf)*deltaT) - d)/(u-d);
lattice = zeros(N+1,N+1);
for j = 0:N
    lattice(N+1,j+1) = max(50*(S0-X(N))/S0*(u^j)*(d^(N-
j)),last)
end
for i = N-1:-1:0
    for j = 0:i
        lattice(i+1,j+1) = max(50*(S0-
X(i+1))/S0*(u^j)*(d^(i-j)),exp(-
r*deltaT)*(p*lattice(i+2,j+2)+(1-p)*lattice(i+2,j+1)))
    end
end
end

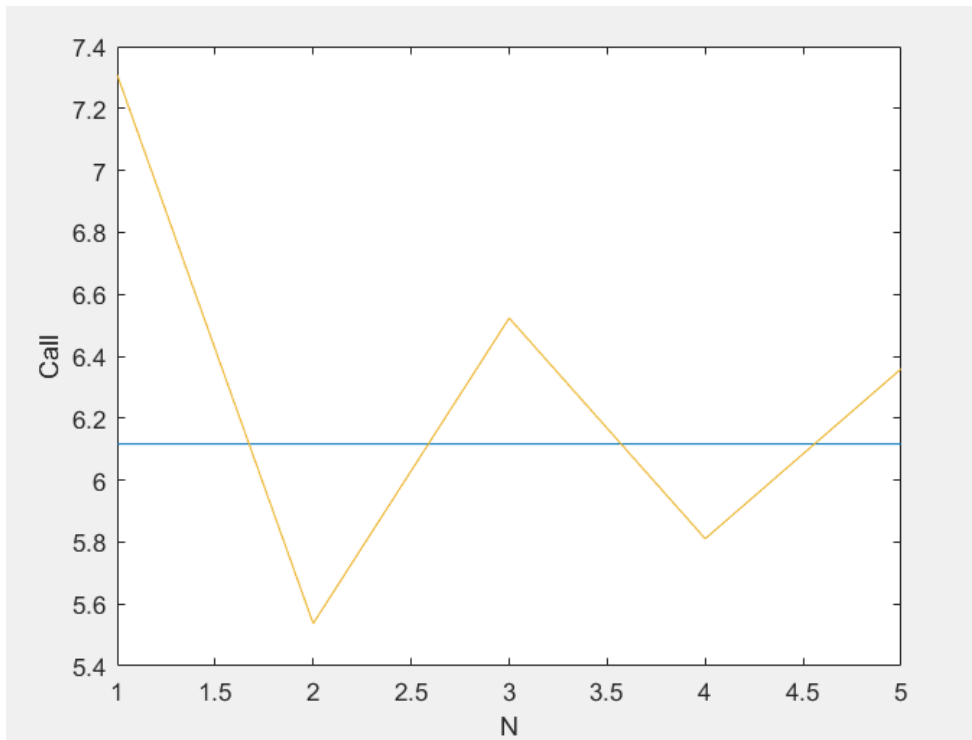
```

```

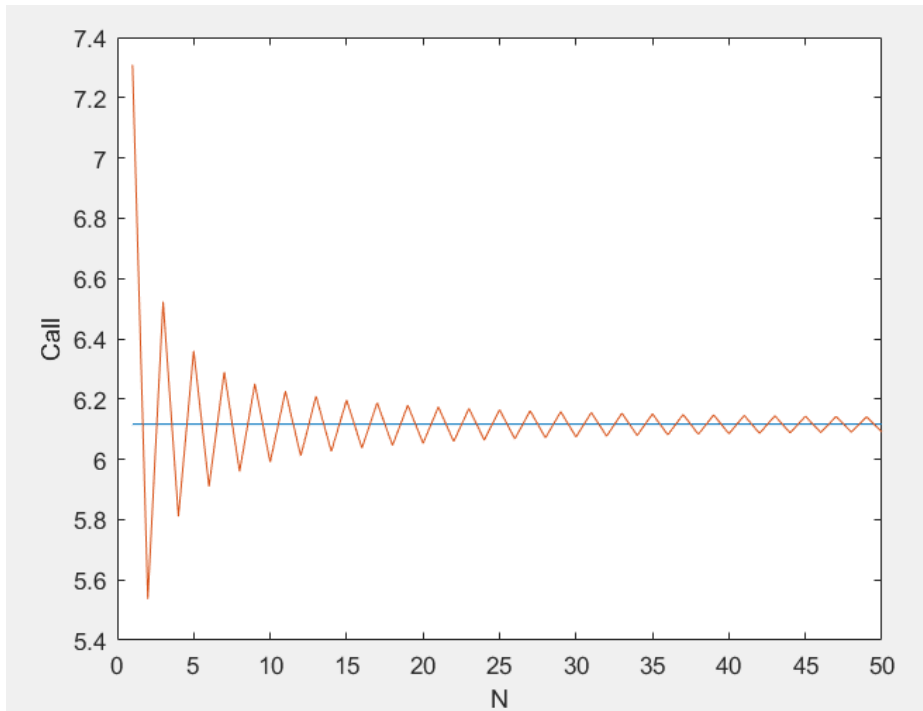
price = lattice(1,1);
end

%AmCallVSEurCall
%input
S0 = 50;
X = 50;
r = 0.1;
sigma = 0.4;
T = 5/12; %for  $\alpha \ll 0$ 
N = 5;
q = 0.1;
%BLS
[BlsC,BlsP] = blsprice(S0,X,r,T,sigma);
LatticeAmC = zeros(1,N);
LatticeC = zeros(1,N);
for i = (1:N)
    LatticeAmC(i) = LatticeAmCall(S0,X,r,T,sigma,i);
    LatticeC(i) = LatticeEurCall(S0,X,r,T,sigma,i);
end
plot(1:N, ones(1,N)*BlsC);
hold on;
plot(1:N,LatticeAmC);
hold on;
plot(1:N, LatticeC);
xlabel('N');
ylabel('Call');

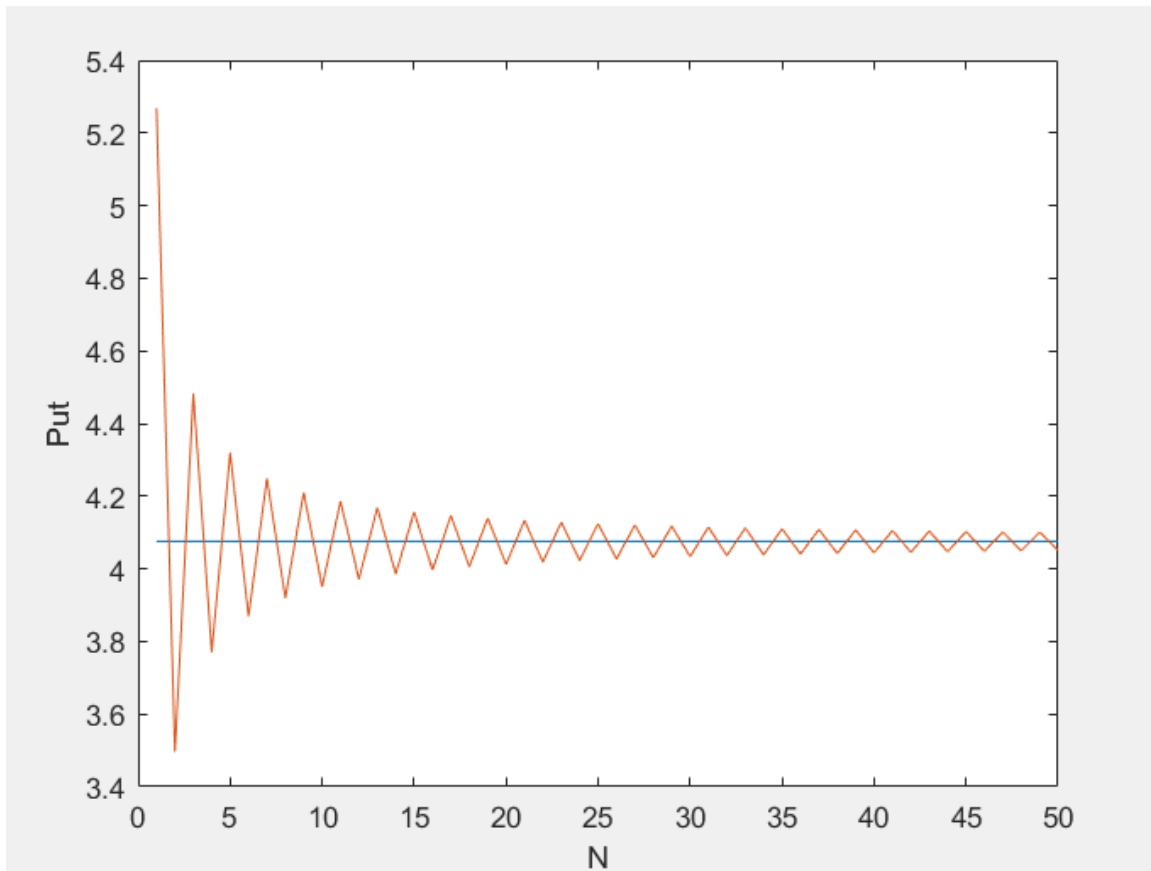
```



```
%LatticeAmCall_VS_BLS
%input
S0 = 50;
X = 50;
r = 0.1;
sigma = 0.4;
T = 5/12; %for  $\sigma \ll 1$ 
N = 50;
%BLS
[BlsC,BlsP] = blsprice(S0,X,r,T,sigma);
LatticeC = zeros(1,N);
for i = (1:N)
    LatticeC(i) = LatticeAmCall(S0,X,r,T,sigma,i);
end
plot(1:N, ones(1,N)*BlsC);
hold on;
plot(1:N, LatticeC);
xlabel('N');
ylabel('Call');
```



```
%CompLatticeBLSPut.m
%input
S0 = 50;
X = 50;
r = 0.1;
sigma = 0.4;
T = 5/12; %for  $\sigma \ll 1$ 
N = 50;
[BlsC,BlsP] = blsprice(S0,X,r,T,sigma);
LatticeP = zeros(1,N);
for i = (1:N)
    LatticeP(i) = LatticeEurPut(S0,X,r,T,sigma,i);
end
plot(1:N, ones(1,N)*BlsP);
hold on;
plot(1:N, LatticeP);
xlabel('N');
ylabel('Put');
```



```
%CompLatticeBLSCall.m
%input
S0 = 50;
X = 50;
r = 0.1;
sigma = 0.4;
T = 5/12; %for 内建
N = 50;
BlsC = blsprice(S0,X,r,T,sigma);
LatticeC = zeros(1,N);
for i = (1:N)
    LatticeC(i) = LatticeEurCall(S0,X,r,T,sigma,i);
end
plot(1:N, ones(1,N)*BlsC);
hold on;
plot(1:N, LatticeC);
xlabel('N');
ylabel('Call');
```

