mathematics

MDPI

Article

Supplementary Materials: Volatility timing: Pricing Barrier Options on DAX XETRA Index

Carlos Esparcia, Elena Ibañez and Francisco Jareño

Matlab codes for option pricing

Modified Black-Scholes model. Historical volatility

```
%% INPUTS
load DAX.mat; % Underlying DAX prices downloaded from Yahoo Finance
r=0.00117; % risk-free rate, previously computed in R-Studio
q=0 % zero dividend yiedl
sigma=0.1372369; % Historical volatility previously computed in R-Studio
k = 10000; % strike and barrier level
H = k; % depends on the option: H= 10000; 10200; 10400; 10600; 10800
x=1;

%% Black-Scholes modified model
prima=zeros(21,1);
for z=1:x:21 %number of steps
S0 = DAX(z);
t1=datetime(datestr(fechas(end)),'Format','dd-MMM-yyyy');
t2=datetime(datestr(fechas(z)),'Format','dd-MMM-yyyy');
T = daysact(t2,t1)/365;   %vencimiento
%% THE MODEL
d1 = (log(S0/k)+(r-q+0.5*(sigma^2))*T)/(sigma*sqrt(T));
d2 = d1-sigma*sqrt(T);
Nd1 = normcdf(d1);
Nd2 = normcdf(d2);
lambda = (r-q+0.5*(sigma^2))/(sigma^2);
x1=(log(S0/H)/(sigma*sqrt(T)))+(lambda*sigma*sqrt(T));
y1 = (log(H/S0)/(sigma*sqrt(T)))+(lambda*sigma*sqrt(T));
Nx1 = normcdf(x1);
Nx1_vol = normcdf(x1-sigma*sqrt(T));
Ny1 = normcdf(y1);
Ny1_vol = normcdf(y1-sigma*sqrt(T));

cdo = (S0*Nx1*exp(-q*T))-((k*exp(-r*T))*Nx1_vol)-((S0*exp(-q*T))*((H/S0)^(2*lambda))*Ny1)+((k*exp(-
r*T))*((H/S0)^(2*lambda-2))*Ny1_vol)

%% current option premium
prima(z,1)=cdo/1000
end
```

Trinomial model. Historical volatility

```
%% INPUTS
load DAX.mat; % Underlying DAX prices downloaded from Yahoo Finance
r=0.00117; % risk-free rate, previously computed in R-Studio
q=0
sigma=0.1372369; % Historical volatility previously computed in R-Studio
k = 10000; % strike and barrier level
H = k; % depends on the option: H= 10000; 10200; 10400; 10600; 10800
n = 89;
```

```
%% TRINOMIAL MODEL
prima=zeros(21,1);
for z=1:1:21 %number of steps
S0 = DAX(z);
t1=datetime(datestr(fechas(end)),'Format','dd-MMM-yyyy');
t2=datetime(datestr(fechas(z)),'Format','dd-MMM-yyyy');
T = daysact(t2,t1)/365;   %maturity
n = n-1;
dt=T/n;
u= exp(sigma*sqrt(2*dt));
d=1/u;
pu=-((r-q-(sigma^2)/2)*dt)/(2*log(u))+((sigma^2)*dt/(2*((log(u))^2)))
pd=((r-q-(sigma^2)/2)*dt)/(2*log(u))+((sigma^2)*dt/(2*((log(u))^2)))
pm= 1-((sigma^2)*dt/(log(u))^2)
sump= pu+pd+pm
fd= exp(-r*dt);
% Asset prices at maturity (period M)
S = zeros(n*2+1,n+1);
x=(n*2+1)/2+0.5;
S(x,:)=S0;

%% UNDERLYING TREE PROCESS

% up underlying tree process
for i = x-1:-1:1
for j=2:1:n+1
S(i,j) = S(i+1,j-1)*u;
end
end

% down underlying tree process
for i = x+1:1:n*2+1
for j=2:1:n+1
S(i,j) = S(i-1,j-1)*d;
end
end


%% PAYOFF
payoff = max(S(:,end) - k, 0).*all(S(:,end)>H,2);
CallA=zeros(n*2+1,n+1)
CallA(:,n+1)=payoff

%% OPTION TREE PROCESS
aux=0;
for j = n:-1:1 %Step back through the tree
for i = aux:(n*2-2-aux)
    if S(i+2,j)>H
    CallA(i+2,j)= fd*(pu*CallA(i+1,j+1) + pm*CallA(i+2,j+1)+pd*CallA(i+3,j+1));
    else
    CallA(i+2,j)= 0;
    end
end
    aux=aux+1;
end

S
```

```
CallA
%% CURRENT OPTION PREMIUM
prima(z,1)=CallA(x,1)/1000
end
```

Trinomial model. GARCH (1,1) volatility

```
%% INPUTS
load sigma.mat; % GARCH(1,1)volatility previously computed in R-Studio
load DAX.mat; % Underlying DAX prices downloaded from Yahoo Finance
r=0.00117; % risk-free rate, previously computed in R-Studio
q=0;
k = 10000; % strike and barrier level
H = k; % depends on the option: H= 10000; 10200; 10400; 10600; 10800
n = 89;

%% TRINOMIAL MODEL
prima=zeros(21,1);
for z=1:1:21 %number of steps
S0 = DAX(z);
t1=datetime(datestr(fechas(end)),'Format','dd-MMM-yyyy');
t2=datetime(datestr(fechas(z)),'Format','dd-MMM-yyyy');
T = daysact(t2,t1)/365;   %maturity
n = n-1;
dt=T/n;
sigma = sigma((end-n+1):end);
u= exp(sigma*sqrt(2*dt));
d=1./u;
pu=-((r*ones(size(sigma))-q-(sigma.^2)/2)*dt)./(2*log(u))+((sigma.^2)*dt./(2*((log(u)).^2)))
pd=((r*ones(size(sigma))-q-(sigma.^2)/2)*dt)./(2*log(u))+((sigma.^2)*dt./(2*((log(u)).^2)))
pm= ones-((sigma.^2)*dt./(log(u)).^2)
sump= pu+pd+pm
fd= exp(-r*dt);
% Asset prices at maturity (period M)
S = zeros(n*2+1,n+1);
x=(n*2+1)/2+0.5;
S(x,:)=S0;

%% UNDERLYING TREE PROCESS

% up underlying tree process
for i = x-1:-1:1
for j=2:1:n+1
S(i,j) = S(i+1,j-1)*u(j-1);
end
end

% down underlying tree process
for i = x+1:1:n*2+1
for j=2:1:n+1
S(i,j) = S(i-1,j-1)*d(j-1);
end
end

%% OPTION PAYOFF
payoff = max(S(:,end) - k, 0).*all(S(:,end)>H,2);
CallA=zeros(n*2+1,n+1)
CallA(:,n+1)=payoff
```

```
%% OPTION TREE PROCESS
aux=0;
for j = n:-1:1 %Step back through the tree
for i = aux:(n*2-2-aux)
    if S(i+2,j)>H
    CallA(i+2,j)= fd*(pu(j)*CallA(i+1,j+1) + pm(j)*CallA(i+2,j+1)+pd(j)*CallA(i+3,j+1));
    else
    CallA(i+2,j)= 0;
    end
end
    aux=aux+1;
end

S
CallA
%% CURRENT OPTION PREMIUM
prima(z,1)=CallA(x,1)/1000
end
```

Binomial model. Historical volatility

```
%% INPUTS
load DAX.mat; % Underlying DAX prices downloaded from Yahoo Finance
r=0.00117; % risk-free rate, previously computed in R-Studio
q=0 % dividend yield
sigma=0.1372369; % Historical volatility previously computed in R-Studio
k = 10000; % strike and barrier level
H = k; % depends on the option: H= 10000; 10200; 10400; 10600; 10800
n = 89;   % number of steps for the tree
x=1;

%% BINOMIAL MODEL
prima=zeros(21,1);
for z=1:x:21 %number of steps
S0 = DAX(z);
t1=datetime(datestr(fechas(end)),'Format','dd-MMM-yyyy');
t2=datetime(datestr(fechas(z)),'Format','dd-MMM-yyyy');
T = daysact(t2,t1)/365;   %maturity
n = n-1;
dt=T/n;
a= exp((r-q)*dt);
u= exp(sigma*sqrt(dt));
d=1/u;
pu= (a-d)/(u-d);
pd= 1-pu;
fd= exp(-r*dt);
% Asset prices at maturity (period M)
S = zeros(n+1,n+1);
S(1,1)=S0;

%% UNDERLYING TREE PROCESS
for y=2:n+1
S(1,y) = S0*u^(y-1);
end

for j=n+1:-1:2

for i = 1:(j-1)
S(1+ i,j) = S(i,j) * d / u; % we build the last column of the tree
```

```
end
end

%% OPTION PAYOFF
payoff = max(S(:,end) - k, 0).*all(S(:,end)>H,2);
CallA=zeros(n+1,n+1)
CallA(:,n+1)=payoff;

%% OPTION TREE PROCESS
for i = n:-1:1 %Step back through the tree
for j = 0:(i-1)
    if S(j+1,i)>H
CallA(j+1,i)= (fd*(pu*CallA(j+1,i+1) + pd*CallA(j+2,i+1)));
    else
CallA(j+1,i)= 0;
    end
end
end

S
CallA
%% CURRENT OPTION PREMIUM
prima(z,1)=CallA(1,1)/1000
end
```

Binomial model. GARCH (1,1) volatility

```
%% INPUTS
load sigma.mat; % GARCH(1,1)volatility previously computed in R-Studio
load DAX.mat; % Underlying DAX prices downloaded from Yahoo Finance

r=0.00117; % risk-free rate, previously computed in R-Studio
q=0;
k = 10000; % strike and barrier level
H = k; % depends on the option: H= 10000; 10200; 10400; 10600; 10800
n = 89;
x=1;

%% BINOMIAL MODEL
prima=zeros(21,1);
for z=1:x:21 %number of steps
S0 = DAX(z);
% S0 = 11401.96973; %Precio de la acción
t1=datetime(datestr(fechas(end)),'Format','dd-MMM-yyyy');
t2=datetime(datestr(fechas(z)),'Format','dd-MMM-yyyy');
T = daysact(t2,t1)/365;   %maturity
n = n-1;
dt=T/n;
sigma = sigma((end-n+1):end);
a= exp((r-q)*dt);
u= exp(sigma*sqrt(dt));
d=1./u;
pu = (a-d)./(u-d);
pd = 1-pu;
total = pu+pd;
fd= exp(-r*dt);
% Asset prices at maturity (period M)
S = zeros(n+1,n+1);
S(1,1)=S0;
```

```
%% UNDERLYING TREE PROCESS
for y=2:n+1
S(1,y) = S0*u(y-1)^(y-1);
end

for j=n+1:-1:2

for i = 1:(j-1)
S(1+ i,j) = S(i,j) * d(j-1) / u(j-1); % we build from the last column of the tree
end
end

%% OPTION PAYOFF
payoff = max(S(:,end) - k, 0).*all(S(:,end)>H,2);
CallA=zeros(n+1,n+1)
CallA(:,n+1)=payoff;

%% OPTION TREE PROCESS
for i = n:-1:1 %Step back through the tree
for j = 0:(i-1)
    if S(j+1,i)>H
CallA(j+1,i)= (fd*(pu(i)*CallA(j+1,i+1) + pd(i)*CallA(j+2,i+1)));
    else
CallA(j+1,i)= 0;
    end
end
end

S
CallA
%% CURRENT OPTION PREMIUM
prima(z,1)=CallA(1,1)/1000
end
```