Appendix

```
%AssetPaths.m
function SPaths=AssetPaths(S0,mu,sigma,T,NSteps,NRepl)
SPaths = zeros(NRepl, 1+NSteps);
SPaths(:,1) = S0;
dt = T/NSteps;
nudt = (mu-0.5*sigma^2)*dt;
sidt = sigma*sqrt(dt);
for i=1:NRepl
  for j=1:NSteps
    SPaths(i,j+1)=SPaths(i,j)*exp(nudt + sidt*randn);
  end
end

%AssetPathsAV.m
function [SPathsA, SPathsB] = AssetPathsAV(S0,mu,sigma,T,NSteps,NRepl)
SPathsA = zeros(NRepl, 1+NSteps);
SPathsA(:,1) = S0;
SPathsB = zeros(NRepl, 1+NSteps);
SPathsB(:,1) = S0;
dt = T/NSteps;
nudt = (mu-0.5*sigma^2)*dt;
sidt = sigma*sqrt(dt);
for i=1:NRepl
   for j=1:NSteps
      SPathsA(i,j+1) = SPathsA(i,j)*exp(nudt + sidt*randn);
      SPathsB(i,j+1) = SPathsB(i,j)*exp(nudt + sidt*randn);
   end
end
```

```
% DownOutPut.m
function P = DOPut(S0,K,r,T,sigma,Sb)
a = (Sb/S0)^(-1 + (2*r / sigma^2));
b = (Sb/S0)^(1 + (2*r / sigma^2));
d1 = (log(S0/K) + (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d2 = (log(S0/K) + (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d3 = (log(S0/Sb) + (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d4 = (log(S0/Sb) + (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d5 = (log(S0/Sb) - (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d6 = (log(S0/Sb) - (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
d7 = (log(S0*K/Sb^2) - (r-sigma^2 / 2)* T) / (sigma*sqrt(T));
d8 = (log(S0*K/Sb^2) - (r+sigma^2 / 2)* T) / (sigma*sqrt(T));
P = K*exp(-r*T)*(normcdf(d4)-normcdf(d2) - ...
    a*(normcdf(d7)-normcdf(d5))) ...
    - S0*(normcdf(d3)-normcdf(d1) - ...
b*(normcdf(d8)-normcdf(d6)));

%DownOutPutMC.m
function [P,CI,NCrossed] = DOPutMC(S0,K,r,T,sigma,Sb,NSteps,NRepl)
[Call,Put] = blsprice(S0,K,r,T,sigma);
Payoff = zeros(NRepl,1);
NCrossed = 0;
for i=1:NRepl
   Path=AssetPaths1(S0,r,sigma,T,NSteps,1);
   crossed = any(Path <= Sb);
   if crossed == 0
      Payoff(i) = max(0, K - Path(NSteps+1));
   else
      Payoff(i) = 0;
      NCrossed = NCrossed + 1;
   end
```

```
end
[P,aux,CI] = normfit( exp(-r*T) * Payoff);

%AVDOPutMC.m
function [P,std,CI] = AVDOPutMC(S0,Sb,K,r,T,sigma,NSteps,NRepl)
[Call,Put] = blsprice(S0,K,r,T,sigma);
PayoffA = zeros(NRepl,1);
PayoffB = zeros(NRepl,1);
for i=1:NRepl
   [PathA, PathB] = AssetPathsAV(S0,r,sigma,T,NSteps,1);
   crossedA = any(PathA <= Sb);
   crossedB = any(PathB <= Sb);
   if crossedA == 0
      PayoffA(i) = max(0,K-PathA(NSteps+1));
   end
   if crossedB == 0
      PayoffB(i) = max(0,K-PathB(NSteps+1));
   end
end
Payoff = (PayoffA + PayoffB)./2;
[P, std, CI] = normfit(exp(-r*T)*Payoff);

%CVDOPutMC.m
function [P,std,CI] = CVDOPutMC(S0,Sb,K,r,T,sigma,NSteps,NRepl,NPilot)
Payoff = zeros(NPilot,1);
VanillaPayoff = zeros(NPilot,1);
[aux, muVanilla] = blsprice(S0,K,r,T,sigma);
for i=1:NPilot
   Path = AssetPaths(S0,r,sigma,T,NSteps,1);
   VanillaPayoff(i) = max(0,K-Path(NSteps+1));
   crossed = any(Path <= Sb);
```

```
   if crossed == 0
   Payoff(i) = max(0,K-Path(NSteps+1));
   end
end
VanillaPayoff = exp(-r*T)*VanillaPayoff;
Payoff = exp(-r*T)*Payoff;

covMat = cov(VanillaPayoff, Payoff);
varVanilla = var(VanillaPayoff);
c = -covMat(1,2)/varVanilla;

newPayoff = zeros(NRepl,1);
newVanillaPayoff = zeros(NRepl,1);
for i=1:NRepl
   Path = AssetPaths(S0,r,sigma,T,NSteps,1);
   newVanillaPayoff(i) = max(0,K-Path(NSteps+1));
   crossed = any(Path <= Sb);
   if crossed == 0
      newPayoff(i) = max(0,K-Path(NSteps+1));
   end
end
newVanillaPayoff = exp(-r*T)*newVanillaPayoff;
newPayoff = exp(-r*T)*newPayoff;
CVpayoff = newPayoff + c*(newVanillaPayoff - muVanilla);
[P, std, CI] = normfit(CVpayoff);

%DOPutMCCond.m
function [Pdo,CI,NCrossed] = ...
  DOPutMCCond(S0,K,r,T,sigma,Sb,NSteps,NRepl)
dt = T/NSteps;
[Call,Put] = blsprice(S0,K,r,T,sigma);
```

```
NCrossed = 0;
Payoff = zeros(NRepl,1);
Times = zeros(NRepl,1);
StockVals = zeros(NRepl,1);
for i=1:NRepl
  Path=AssetPaths1(S0,r,sigma,T,NSteps,1);
  tcrossed = min(find( Path <= Sb ));
  if not(isempty(tcrossed))
    NCrossed = NCrossed + 1;
                Times(NCrossed) = (tcrossed-1) * dt;
                StockVals(NCrossed) = Path(tcrossed);
  end
end
if (NCrossed > 0)
  [Caux, Paux] = blsprice(StockVals(1:NCrossed),K,r,...
    T-Times(1:NCrossed),sigma);
  Payoff(1:NCrossed) = exp(-r*Times(1:NCrossed)) .* Paux;
end
[Pdo, aux, CI] = normfit(Put - Payoff);

%DOPutMCCondIS.m
function [Pdo,CI,NCrossed] = ...
  DOPutMCCondIS(S0,K,r,T,sigma,Sb,NSteps,NRepl,bp)
dt = T/NSteps;
nudt = (r-0.5*sigma^2)*dt;
b = bp*nudt;
sidt = sigma*sqrt(dt);
[Call,Put] = blsprice(S0,K,r,T,sigma);

NCrossed = 0;
Payoff = zeros(NRepl,1);
```

```
Times = zeros(NRepl,1);
StockVals = zeros(NRepl,1);
ISRatio = zeros(NRepl,1);
for i=1:NRepl
        vetZ = nudt - b + sidt*randn(1,NSteps);
        LogPath = cumsum([log(S0), vetZ]);
        Path = exp(LogPath);
  jcrossed = min(find( Path <= Sb ));
  if not(isempty(jcrossed))
    NCrossed = NCrossed + 1;
    TBreach = jcrossed - 1;
                Times(NCrossed) = TBreach * dt;
                StockVals(NCrossed) = Path(jcrossed);
    ISRatio(NCrossed) = exp( TBreach*b^2/2/sigma^2/dt +...
      b/sigma^2/dt*sum(vetZ(1:TBreach)) - ...
      TBreach*b/sigma^2*(r - sigma^2/2));
  end
end
if (NCrossed > 0)
  [Caux, Paux] = blsprice(StockVals(1:NCrossed),K,r,...
    T-Times(1:NCrossed),sigma);
  Payoff(1:NCrossed) = exp(-r*Times(1:NCrossed)) .* Paux ...
    .* ISRatio(1:NCrossed);
end
[Pdo, aux, CI] = normfit(Put - Payoff);
```