```
% CDO Pricing
% First to default basket
% MonteCarlo Simulation
% by Antonio Rivela 2009



% Initial Data for the simulation


function result = CDOPricing(N,R,lambdaf,rho,n,c,a,d,r,No,flag)
    TM = 5;                    % length of the CDO
    tstep = 0.5;               % the coupon payments
    nn  = n * N;               % the total notional
    loss = n * ( 1 - R);       % the total loss
    T=0:tstep:TM;              % vector for the fixed coupon dates
    Tmod=repmat(T,N,1);        % matrix of fixed coupon dates for all N companies
    discount=exp(-r*T(2:end)); % discounted fixed coupon vector
    randn('state',0);          % just to initialize the generator
    MRho = repmat(rho,N,N);    % initializing the correlation matrix
    for i=1:N
       MRho(i,i) = 1;          % filling diagonal entries with 1
    end
    MRho = chol(MRho)';        % doing the Cholesky factorization
    fixedtot = 0;              % initializing for fixed leg total
    floattot = 0;              % initializing for floating leg total
    sqfixtot = 0;              % for standard error estimate
    sqfltot  = 0;              % for standard error estimate
    PMat = randn(N, No);       % initialzing the Gaussian matrix
    PMat1 = MRho * PMat;       % to get the correlated Gaussian matrix
    PMat11 = normcdf(PMat1,0,1); % take the CDF to make them a copula
    PMat2 = -log(1 - PMat11)/ lambdaf;    % inverse function to get the default time
    for i=1:No                 % loop for different paths of MC
       PMat3=PMat2(:,i);       % getting the i'th path
       Pmatmod=repmat(PMat3,1,2*TM+1);   % getting it for the fixed coupon dates
       Temp1= Pmatmod<Tmod;    % keeping default times that are only within the CDS maturity
       Lmat = loss*Temp1;      % getting the losses matrix by multiplying with the defaults
```

```
    Tloss= sum(Lmat);               % summing up the losses
    Ploss= Tloss/nn;                % getting the loss percentages
    Plossum =max(Ploss-a,0)-max(Ploss-d,0);    % getting the loss percentage in the tranche
    Lossum =Plossum*nn;                    % getting the absolute loss in the tranche
    tempplos =Plossum(2:end)-Plossum(1:end-1); %
    temp2 =nn * tempplos;
    temp =nn*(d-a-Plossum);                    % getting the notional left in the tranche
    coupon = tstep*(temp(1:end-1)+ temp(2:end))/2; % getting the fixed coupons
    fl_flows=discount.*temp2;                  % getting the discounted floating flows
    fx_flows=discount.*coupon*c;               % getting the discounted fixed flows
    Vfloat=sum(fl_flows);                      % the total floating flows for this path
    floattot = floattot + Vfloat;              % the total floating flows until now
    sqfltot  = sqfltot  + (Vfloat^2);          % to get the standard error square term
    Vfixed=sum(fx_flows);                      % the total fixed flows for this path
    fixedtot = fixedtot + Vfixed;              % the total fixed flows until now
    sqfixtot = sqfixtot + (Vfixed^2);          % to get the standard error square term
 end
% displaying out the results
disp('Monte Carlo fixed leg value estimate:')
result(1,1) = fixedtot/No
disp('Monte Carlo floating leg value estimate:')
result(1,2) = floattot/No
disp('Monte Carlo CDO Value estimate:')
if flag == 1                               % if flag is set the buyer's value is displayed
    result(1,3) = (floattot - fixedtot)/No
else
    result(1,3) = (fixedtot - floattot)/No                 % if flag is not set the seller's value is displayed
end
disp('Monte Carlo Breakeven Spread estimate:')
result(1,4) = floattot/(fixedtot/c)
disp('Monte Carlo Fixed leg standard error:')
result(1,5) =  (1/No) * sqrt(sqfixtot - ((1/No) * (fixedtot^2)))
disp('Monte Carlo Floating leg standard error:')
result(1,6) = (1/No) * sqrt(sqfltot - ((1/No) * (floattot^2)))
```