

```

% Monte Carlo Simulation
% First to default basket
% Basket of France Telecom, Repsol & Telecom Italia
% by Antonio Rivela 2009
%

% Initial Data for the simulation

function ftd_credit_spread=ftd4(n_simul,corr_mat)

clc

%printf('First of Default Credit Basket.\n')
%printf("Monte Carlo Credit Simulation.\n")
%printf("Antonio Rivela.\n")
%printf("IE Business School Nov 2009.\n")
%printf("-----\n")

n_simul

s=1

s1=0.0085
s2=0.0258
s3=0.0320

recovery=0.30
r=0.03

% corr=0.8046

% corr_mat=[1 corr corr ; corr 1 corr ; corr corr 1]

corr_mat

lambda1=s1/(1-recovery);
lambda2=s2/(1-recovery);
lambda3=s3/(1-recovery);

```

```

% Convention ACT/360 for the spread
s=s*365/360;
% printf("-----\n")

% Simulation of Normal(0,1)

independent_normal=normrnd(0,1,n_simul,3);

% Simulation of Normal(0,1) correlated – Cholesky Transformation

correlated_normal=(chol(corr_mat)*independent_normal);

% F normal of the Normal correlated samples

normal_F=normcdf(correlated_normal,0,1)';

% Exponential inverse T of the F normal
% Simulation of times to default

times_exp_1=expinv(normal_F(:,1),1/lambda1);
times_exp_2=expinv(normal_F(:,2),1/lambda2);
times_exp_3=expinv(normal_F(:,3),1/lambda3);

times_exp=min(min(times_exp_1,times_exp_2),times_exp_3);

% Average Time of the simulation

average_time_simulation=mean(times_exp);

% Mask that contains a 1 if credit survived or a zero if it did not survive

ones=(times_exp<=5);

```

```
% Calculate the present value of each of the losses
```

```
losses=exp(-times_exp.*r).*(1-recovery).*ones;
```

```
% Calculate the sum of all the present valued Losses. Right Side of Equation
```

```
% printf("Calculating Monte Carlo paths...\n")
```

```
loss_pv_montecarlo=mean(losses);
```

```
% Build a vector that will contain the spreads of the CDS transaction
```

```
s_vector=zeros(n_simul,5);
```

```
% Build a vector with the times of each spread of the CDS transaction
```

```
s_time_vector=[1 2 3 4 5];
```

```
% calculate the discount factor of each of the spreads of the CDS transaction with discount rate r
```

```
df_s_time_vector=exp(-s_time_vector*r);
```

```
% PV spreads = Multiply discount factors by spreads of the CDS transaction
```

```
s_vector(:,1)=s*df_s_time_vector(1);
```

```
s_vector(:,2)=s*df_s_time_vector(2);
```

```
s_vector(:,3)=s*df_s_time_vector(3);
```

```
s_vector(:,4)=s*df_s_time_vector(4);
```

```
s_vector(:,5)=s*df_s_time_vector(5);
```

```
% Mask that takes into account as well the fact that if the credit has defaulted
```

```
% then you stop making the "s" payments
```

```
s_mask=[times_exp>1 times_exp>2 times_exp>3 times_exp>4 times_exp>5];
```

```
s_vector_after_mask=s_vector.*s_mask;
```

```
% losses_spreads_montecarlo=sum(mean(s_vector_after_mask))
```

```
% Losses_spreads_montecarlo2 considers an accrued coupon on the credit spread if a default occurs
```

```
losses_spreads_montecarlo2=s_vector_after_mask(:,1)+s_vector_after_mask(:,2)+s_vector_after_mask(:,3)+s_vector_after_mask(:,4)+s_vector_after_mask(:,5);
```

```
accrued_spread_vector=s*(times_exp-floor(times_exp)).*exp(-r*times_exp).*(times_exp<5);
```

```
losses_spreads_montecarlo2=mean(losses_spreads_montecarlo2+accrued_spread_vector);
```

```
ftd_credit_spread=loss_pv_montecarlo/losses_spreads_montecarlo2;
```

```
end
```