Case Study | fairmat

Autocallable bond

Revision #1

1 Introduction

An *autocallable bond* is a structured product which offers the opportunity for an *early redemption if a predefined event occurs* and pays *coupons* conditioned to the realization of other events. Both these opportunities are linked to a function P of the performance of the underlying which may be composed by several stocks. In our case the controlling function is the worst of the performances in a basket of assets, at all the observation dates $t = 1, 2, \ldots, T$:

$$P_{\mathrm{t}} = \min_{\forall i \in \{1,2,\ldots,K\}} \left\{ \frac{I_{\mathrm{i,t}}}{I_{\mathrm{i,0}}} \right\},$$

where K is the number of assets.

A peculiar characteristic of the *autocallable bond* is the presence of an *early redemption* clause (hereinafter called *Early Redemption Event* or *ERE*) which, however, is not controlled by the issuer (the so-called *callability*), but rather, is automatically carried when P at certain time t exceeds a certain threshold (hereinafter called *High Trigger Level* or *HTL*).

In general, the contract provides alternative scenarios even in the case in which an *ERE* does not occur. For example, if the performance of the underlying reaches a minimum threshold (hereinafter called *Low Trigger Level* or *LTL*), the coupon will be a fixed rate (hereinafter called *fix*).

Often the *ERE* is associated with a higher coupon than that normally is paid under the assumption of continuation of the contract. This coupon is in usually a multiple of *fix* (*gear*). We can summarize the payoff formula π, at all observation dates $t = 1, 2, \ldots, T$, as follows:

$$\pi_t = \begin{cases} 0, & \text{if} \quad P_{\mathrm{t}} < LTL \\ fix, & \text{if} \quad LTL \leq P_{\mathrm{t}} < HTL \\ gear \times fix + Principal, & \text{if} \quad P_{\mathrm{t}} \geq HTL \\ & \text{(an } ERE \text{ occured).} \end{cases}$$

Autocallable Example		
Principal	100	
Trade Date	12/05/2011	
Effective Date (t_0)	14/05/2011	
Termination Date	14/05/2016	
Issuer payment frequency	Annual	in the case of ERE does not happen
	At certain Observation/Payment Dates (see Table 2)	in the case of ERE happens
	At Termination Date	in the case of ERE doesn't happen until Termination Date
Exchange		**Issuer**
At Observation/Payment Dates_t ($t = 1, 2, 3, 4$)	If $P_t < 100\%$	0.00%
	If $100\% \leq P_t < 120\%$	5.00%
	If $P_t \geq 120\%$	$[2 \times 5.00\% + \text{Principal}]^*$
	* If ERE occurs, the contract pays the payoff at time t and, after, ceases to have effect.	
At Termination Date ($t = 5$)	Only if an ERE_t ($t = 1, 2, 3, 4$) is never occured:	
	If $P_t < \text{LTL}$	$0.00\% + \text{Principal}$
	If $\text{LTL} \leq P_t < \text{HTL}$	$5.00\% + \text{Principal}$
	If $P_t \geq \text{HTL}$	$2 * 5.00\% + \text{Principal}$
	$P_t =$ is the portfolio performance, at time t, described as $P_t = min[I_{i,t}/I_{i,0}]$ with $i = 1, 2, 3$ number of indexes $t = 1, 2, 3, 4, 5$ number of observation/payment dates and $I_{i,t}$ = close value of i-th Index at time t $I_{i,0}$ = close value of i-th Index at time t_0	
	ERE_t = Early Redemption Event at time t. LTL = Low Trigger Level (100%) HTL = High Trigger Level (120%)	
Conventions		**Issuer**
Day Count Convention	*Following*	
Day Count Fraction	*Act*/360, Unadjusted	

Table 1: *Autocallable bond* case study.

2 An example

In this *autocallable bond* case study we have assumed that the controlling function is the worst performance in a basket of three assets among three different financial markets: the European FTSE MIB Index, the American Standard&Poor 500 Index and the Japanese Nikkei-225 Index.[1]

The main features of the *autocallable bond* case study are shown in Table 1 and the Observation/Payment Dates are reported in Table 2.

The coupon of the bond is paid once a year for the 5 years duration of the contract and is defined by the value of the function P_t.

If P_t is below a *Low Trigger Level* (LTL, equals to 100%), the coupon is

[1] The reference to indices listed in different currencies (USA Dollar, Japanese yen), for cash flows into domestic currency (Euro), implies a "quanto" adjustment. This is automatically computed by *Fairmat*.

t	Observation/Payment Dates
1	14/05/2012
2	14/05/2013
3	14/05/2014
4	14/05/2015
5	14/05/2016

Table 2: *Autocallable bond* case study: Observation/Payment Dates. The last date is also the termination dates.

equal to 0%, otherwise it is 5%. In the case where P_t exceeds a *High Trigger Level* (HTL, equals to 120%), an *Early Redemption Event* (ERE) happens: at time t, the coupon bond is $2 \times 5\%$, the notional is reimbursed and the contract shall cease to have effect. This is summarized in the formula below:

$$\pi_t = \begin{cases} 0.00\%, & \text{if} \quad P_t < 100\% \\ 5.00\%, & \text{if} \quad 100\% \leq P_t < 120\% \\ 2 \times 5.00\% + \text{Principal}, & \text{if} \quad P_t \geq 120\% \\ & \text{(an Early Red. Event occured).} \end{cases}$$

for $t = 1, 2, 3, 4$. If no*Early Redemption Event* realizes in years $t = 1, 2, 3, 4$, at Termination Date ($t = 5$) the payoff is the following:

$$\pi_5 = \begin{cases} \text{Principal} + 0.00\%, & \text{if} \quad P_5 < 100\% \\ \text{Principal} + 5.00\%, & \text{if} \quad 100\% \leq P_5 < 120\% \\ \text{Principal} + 2 \times 5.00\%, & \text{if} \quad P_5 \geq 120\%. \end{cases}$$

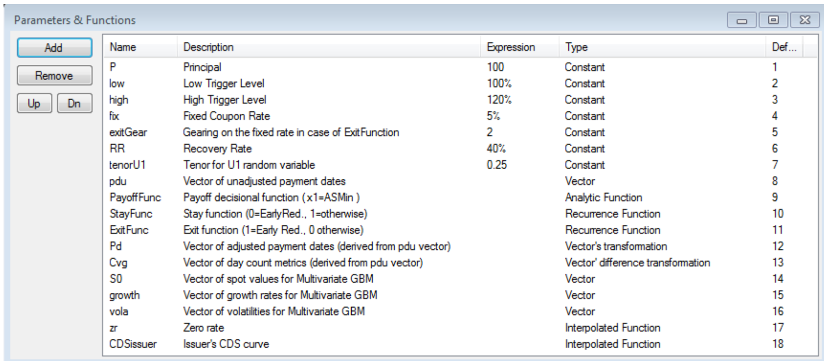
3 An implementation with *Fairmat*

The implementation of the *autocallable bond* described in the previous section through *Fairmat* needs a series of input data, as shown into Figure 1

The Inputs represented in Figure 1 can be categorized into three main classes: the *Contract specific* parameters are described in Table 3, the parameters related to *Market* data as described in Table 4 and finally, *Auxiliary* and *Instrumental* objects (objects and functions that represent transformations of inputs or stochastic process are represented in Table 5.

From a modelling point of view, the most peculiar feature of the *autocallable bond* is the path dependency related to the early redemption permitted in the contract. In Fairmat we model path dependency with the so called *Recurrence Functions* which in general are functions of time, of the stochastic processes and of their previous values.

`StayFunc` recurrence function — shown in Figure 2 — is an indicator function indicating when ERE has not yet happened. If `StayFunc` equals `1` then the contract continues its effects, otherwise `StayFunc` verifies an ERE (equals


Parameters & Functions

Add
Remove
Up Dn

Name	Description	Expression	Type	Def...
P	Principal	100	Constant	1
low	Low Trigger Level	100%	Constant	2
high	High Trigger Level	120%	Constant	3
fix	Fixed Coupon Rate	5%	Constant	4
exitGear	Gearing on the fixed rate in case of ExitFunction	2	Constant	5
RR	Recovery Rate	40%	Constant	6
tenorU1	Tenor for U1 random variable	0.25	Constant	7
pdu	Vector of unadjusted payment dates		Vector	8
PayoffFunc	Payoff decisional function (x1=ASMin)		Analytic Function	9
StayFunc	Stay function (0=EarlyRed., 1=otherwise)		Recurrence Function	10
ExitFunc	Exit function (1=Early Red., 0 otherwise)		Recurrence Function	11
Pd	Vector of adjusted payment dates (derived from pdu vector)		Vector's transformation	12
Cvg	Vector of day count metrics (derived from pdu vector)		Vector' difference transformation	13
S0	Vector of spot values for Multivariate GBM		Vector	14
growth	Vector of growth rates for Multivariate GBM		Vector	15
vola	Vector of volatilities for Multivariate GBM		Vector	16
zr	Zero rate		Interpolated Function	17
CDSissuer	Issuer's CDS curve		Interpolated Function	18

Figure 1: *Autocallable bond*: *Parameters&Functions* section on *Fairmat Professional*.

Name	Type	Description
P	Constant	is the principal, in this case *bullet*
low	Constant	is the *Low Trigger Level*, set at 100%
high	Constant	is the *High Trigger Level*, set at 100%
fix	Constant	is the fixed coupon rate, set at 5%
exitGear	Constant	is the gearing, prefixed to the coupon rate, in case of *Early Redemption Event* or in case the contract comes until maturity (t=5) and $\text{PTFperf}_5 \geq$ High Trigger Level. It is set at 2
pdu	Vector	is the vector of payment dates (unadjusted), size 5×1, used for auxiliary items `Pd` and `Cvg`

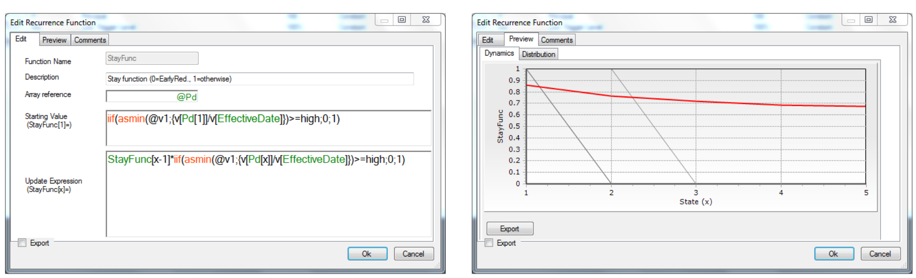
Table 3: *Fairmat Parameters&Functions* enviroment: *Contract specific* parameters.

Name	Type	Description
S0	Vector	is the vector of base values, size 3×1, of the Multivariate GBM process
growth	Vector	is the vector of growth rates, size 3×1, of the Multivariate GBM process
vola	Vector	is the vector of volatilities, size 3×1, of the Multivariate GBM process
zr	Interpolated Function	zero rate (derived from spot rate through a bootstrap method)
CDSgen	Interpolated Function	is the *Credit Default Swap* curve, consists of 8 set maturity points. Each of these points is populated by a spread, in basis point

Table 4: *Fairmat Parameters&Functions* enviroment: *Market* data.

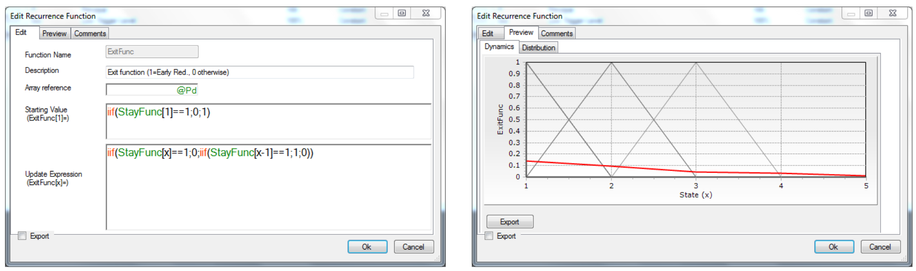
Name	Type	Description
Pd	Vector	is the date's vector transformation, derived from `pdu` vector, with the same size (5×1)
Cvg	Vector	is the date's vector difference transformation, derived from `pdu` vector, with the same size (5×1). For the first element (`Cvg[1]`) uses also the t_0.
PayoffFunc	Analytic Function	is an analytic function which expresses the "floating" coupon (0.00%, 5.00% or $2 \times 5.00\%$) depends on the dynamic of a portfolio performance compared with a minimum-maximum threshold. See below for further information.
StayFunc	Recurrence Function	is a recurrence function used by `PayoffFunc`, indicating that ERE is not yet happened.
ExitFunc	Recurrence Function	is a recurrence function which indicates the date in which ERE happens.

Table 5: *Fairmat Parameters&Functions* section: *Auxiliary* and *Instrumental* variables.



(a) `StayFunc`: *Edit* tab. **(b)** `StayFunc`: *Preview* tab.

Figure 2: `StayFunc` recurrence function on *Fairmat*.



(a) `ExitFunc`: *Edit* tab. **(b)** `ExitFunc`: *Preview* tab.

Figure 3: `ExitFunc` recurrence function on *Fairmat*.

t	1	2	3	4	5
StayFunc	1	1	1	0	0
ExitFunc	0	0	1	0	0

Table 6: One trajectory of `StayFunc` and `ExitFunc` when ERE happens at time $t = 3$.

`0`). The initial value expression of **`StayFunc`** is:

```
iif(asmin(@v1;{v[pd[1]]/v[EffectiveDate]})>=high;0;1)
```

the update expression is defined as follows:

```
StayFunc[x-1]*iif(asmin(@v1;{v[pd[x]]/v[EffectiveDate]})>=high;0;1)
```

In the expression enters the previous value of **`StayFunc`**. The recursion models the fact that when an ERE happens at time t, the contract ceases to have effect for every subsequent date.

The **`ExitFunc`** recurrence function — shown in Figure 3 — is an indicator function which is equal to 1, only at the date in which an ERE happens. **`ExitFunc`** is initialized with the following expression:

```
iif(StayFunc[1]==1;0;1)
```

and the update expression is as follows:

```
iif(StayFunc[x]==1;0;iif(StayFunc[x-1]==1;1;0))
```

In Table 6 is visualized a joint realization of `StayFunc` and `ExitFunc`.

Finally **`Payoff`**, shown in Figure 4, is an analytic function which calculates the *coupon* as function of the the dynamic of portfolio performance.

`StayFunc`, **`ExitFunc`** and **`Payoff`** are used in the *Option Map*, as you see in Figure 6. With more details in the payoff expression reported below:

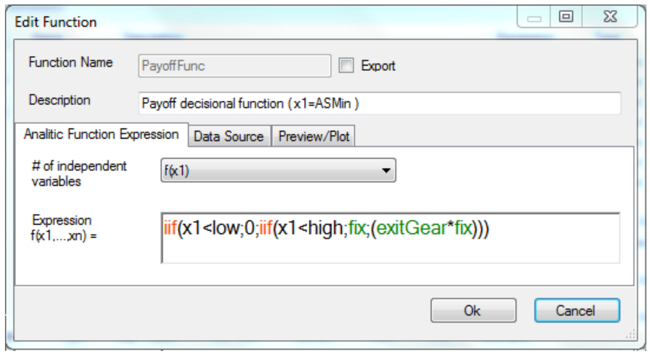
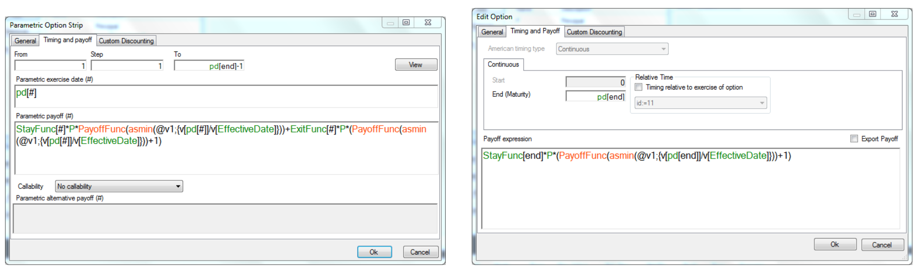


Figure 4: `PayoffFunc` analytic function on *Fairmat*.



(a) *Option Map: strip of options* 1–4. **(b)** *Option Map: custom option* 5.

Figure 5: *Payment legs* on *Fairmat*: strip of options and custom option.

```
StayFunc[\#]*N*ACE(asmin(\@v1;\{v[pd[\#]]/v[EffectiveDate]\}))+
ExitFunc[\#]*N*(ACE(asmin(\@v1;\{v[pd[\#]]/v[EffectiveDate]\}))+1)}
```

The function `ASMin(@basket;{expr})` computes the minimum of an expression inside the brackets, for every element of the `basket`. In detail, for this case study, we are interested in evaluating the worst performance of three indexes. In our case the `basket` is the multivariate GBM process, `v1`, where each component represents the single index, and the expression is **`v[pd[#]]/v[EffectiveDate]`** with **`v`** indicating the generic basket element.

Credit Value adjustment can be calculated in Fairmat by pre-posing a CVA block before the payments blocks.

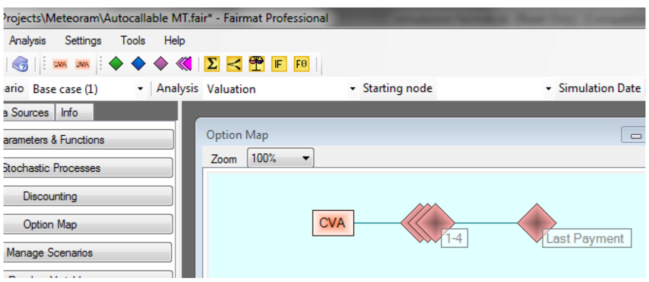


Figure 6: The whole option map including the CVA block.