UNIT 6

BEYOND COPULAS: IMPLIED EXPECTED TRANCHE LOSS AND DYNAMIC LOSS MODELS

Damiano Brigo
www.damianobrigo.it

UNIT 6. Implied Expected Tranche Loss and Dynamic Loss Models

- A model independent paradigm: Implied Expected Tranche Loss;
- Implied ETL: Examples;
- Back to modeling: Bottom Up (Copula) vs Top Down models;
- Dynamic Loss Models: the Generalized Poisson Loss (GPL) model;
- Calibration examples;
- Consistency with single names: Common Poisson Shock framework;
- Avoiding infinite defaults: the Cluster Adjusted GPL model (GPCL);
- Calibration examples;

A model-independent approach: Implied Expected Tranche Loss

Recall: Information contained in CDO quotes. Market quoted spreads for indices and tranches on standardized pools:

$$R_{\text{index}}(0) = \frac{\mathbb{E}_0\left[\int_0^T D(0,u)d\mathsf{Loss}(u)\right]}{\mathbb{E}_0\left[\sum_{i=1}^n \alpha_i D(0,T_i)\left(1-\frac{\mathsf{Loss}(T_i)}{1-\mathsf{Rec}}\right)\right]}$$

$$R_{0,T}^{A,B}(0) = \frac{\mathbb{E}_0\Big[\int_0^T D(0,t)d\mathsf{Loss}_{A,B}^{tr}(t)\Big] - U_{0,T}^{A,B}(0)}{\mathbb{E}_0\Big[\sum_{i=1}^b \alpha_i D(0,T_i)(1-\mathsf{Loss}_{A,B}^{tr}(T_i))\Big]}$$

If these spreads R are the only implied correlation info we have in the market, **we see that the only information we can infer are "expected losses" and "expected tranche losses" (ETL).**

Index and Tranche NPV as a function of ETL

The tranches and the index pay their spread on quarterly dates $T_1, T_2, ..., T_b$, expressed as year fractions (we call the start date $T_0 = 0$). We assume a constant recovery of 40%.

The prices of the premium and default legs of the index can be rewritten as:

$$\begin{aligned} \text{PrPremLeg}_{\text{index}} &= R_{\text{index}} \text{Annuity}_{\text{index}} \\ \text{Annuity}_{\text{ind}} &= \sum_{i=1}^{N} (T_i - T_{i-1}) D(0, T_i) \left(1 - \frac{E[L_i] + E[L_{i-1}]}{2(1 - \text{R}\textsc{ec})}\right) \\ \text{PrDefaultLeg}_{\text{ind}} &= \sum_{i=1}^{N} D(0, T_i) \left[E[L_i] - E[L_{i-1}]\right] \end{aligned} \quad (33)$$

where $L_i = \text{L}\textsc{oss}(T_i)$ and the notation for the index annuity, the index premium leg, the index spread and the index default leg is self evident as before.

Index and Tranche NPV as a function of ETL

The price of the legs of the tranche with attachment A and detachment B is:

$$\mathrm{PrPremLeg}_{\mathrm{A,B}} = \begin{cases} \mathrm{U}_{\mathrm{A,B}} + 0.05 \cdot \mathrm{Annuity}_{\mathrm{A,B}}, & A = 0 \\ R^{\mathrm{A,B}} \mathrm{Annuity}_{\mathrm{A,B}}, & A > 0 \end{cases} \tag{34}$$

$$\mathrm{Annuity}_{\mathrm{A,B}} = \sum_{i=1}^{N} (T_i - T_{i-1}) D(0, T_i) \left(1 - \frac{E[L_i^{A,B}] + E[L_{i-1}^{A,B}]}{2} \right) \tag{35}$$

$$\mathrm{PrDefautLeg}_{\mathrm{A,B}} = \sum_{i=1}^{N} D(0, T_i) \left[E[L_i^{A,B}] - E[L_{i-1}^{A,B}] \right] \tag{36}$$

$$E\left[L_i^{A,B}\right] = \frac{E\left[\max(L_i - A, 0)\right] - E\left[\max(L_i - B, 0)\right]}{B - A} \tag{37}$$

$$= \frac{B}{B-A} E\left[\min(L_i, B)\right] - \frac{A}{B-A} E\left[\min(L_i, A)\right]$$

Index and Tranche NPV as a function of ETL

If we have for a given date the ETL's throughout the entire capital structure (all (A, B)'s) then given the expected recovery we can back-out the expected portfolio total loss, that, divided by $(1 - \text{Rec})$ gives the expected default rate needed for the index valuation.

To span the entire capital structure we need a set of k tranches with attachments A_j and detachments B_j with $j = 1, ..., k$ where $A_1 = 0$, $B_k = 1$ and $A_{i+1} = B_i$. The expected portfolio loss is then the summation of the ETL multiplied by the tranche depth (detachment minus attachment):

$$\sum_{i=1}^{k} L_t^{A_i,B_i}(B_i - A_i) = L_t \quad \Rightarrow \sum_{i=1}^{k} E[L_t^{A_i,B_i}](B_i - A_i) = E[L_t] \qquad (38)$$

Index and Tranche NPV as a function of ETL

The standardized iTraxx tranches have detachments (3%, 6%, 9%, 12%, 22%). To price each of these 5 tranches we need the ETL on all payment dates T_i. To price the index we would also need the ETL of the 22%-100% tranche to complete the expected total loss.

To price the 3y tranches and index (6 market quotes $=$ 5 tranches $+$ 1 index) we will be looking for the 6 unknown 3y ETL's that will set the net present value of the instruments as close as possible to 0.

Index and Tranche NPV as a function of ETL

The ETL on all payment dates with maturity shorter than 3y will be obtained interpolating for each tranche between time 0 (by definition this will be 0) and the 3y unknown ETL to be found.

Once the 3y nodal ETL's matching the data are found, to price the 5 years tranches and index we will need also the ETL between 3 and 5 years. For each tranche this will be obtained by interpolation between the expected tranche loss at 3y and the 6 unknown ETL's at 5y to be found. We then iterate the procedure.

Index and Tranche NPV as a function of ETL

We call $f(t, h, k)$ the ETL at time t of the tranche with attachment h and detachment k (to simplify the notation we will often identify the seniority of the tranche in the capital structure of the CDO only through the detachment point k, writing $f(t, h, k) = f(t, k)$ when h is clear from the context).

Index and Tranche NPV as a function of ETL

	$0\% - 3\%$	$3\% - 6\%$	$6\% - 9\%$	$9\% - 12\%$	$12\% - 22\%$	$22\% - 100\%$
$t = 0$	0	0	0	0	0	0
...						
$t = 3$	$f(3y, 3\%)$	$f(3, 6)$	...			$f(3, 100)$
...						
$t = 5$	$f(5y, 3\%)$	$f(5, 6)$	...			$f(5, 100)$
...						
$t = 7$	$f(7y, 3\%)$	$f(7, 6)$	...			$f(7, 100)$
...						
$t = 10$	$f(10y, 3\%)$	$f(10, 6)$	...			$f(10, 100)$

The set of $4 \cdot 10 \cdot 6 = 240$ f's (one for each quarterly payment date and for each tranche) created by interpolating the $4 \cdot 6 = 24$ basic nodal $f(t, k)$'s (one for each market maturity date and for each tranche) will be used to set the npv of the instruments as close to 0 as possible whilst maintaining the constraints (40) below. For the generic tranche the npv of the premium and default leg expressed in terms of $f(t, k)$ is obtained by Equations (34-36) by substituting

$$E[L^{A,B}{}_i] = f(T_i, A, B).$$

Index and Tranche NPV as a function of ETL

The npv of the premium and default leg of the index expressed in terms of $f(t,k)$ is instead given by (33) where we substitute

$$E[L_i] = \sum_{j=1}^{k} f(T_i, k_j)(k_j - k_{j-1}) \tag{39}$$

(recall that $f(T_i, k_j) = f(T_i, k_{j-1}, k_j)$). We will require that the ETL $f(t,k)$ is non-decreasing in t and non-increasing in k, both requirements being natural given that the loss of a pool should be non-decreasing in time and that the tranched losses re-scaled by the tranche thickness across adjacent attachment-detachment intervals decreases as we move to larger intervals. These requirements ensure also that (39) is increasing in T. Being $f(t,k)$ the expectation of a quantity bounded between 0 and 1, we will also require that $f(t,k)$ lies in the $[0,1]$ interval.

Index and Tranche NPV as a function of ETL

The set of constraints will thus be:

$$\begin{cases} 0 \leq f(t,h,k) \leq 1 \\ f(T_i,h,k) \geq f(T_{i-1},h,k) \\ f(t,k_{j-1},k_j) \leq f(t,k_{j-2},k_{j-1}) \end{cases} \tag{40}$$

We further check a posteriori the condition that the ETL profile for equity tranche losses (attachment-detachment $A-B$ with $A=0$) implicit in our f table be convex with respect to B. Indeed (Breeden Litzenberger) the second derivative of the expected equity tranche loss with respect to the detachment B is related to the opposite of the risk neutral loss density, and as such must be negative.

To find equity ETL remember the link between tranches and equity tranches: Invert

$$f(t,k_{j-1},k_j) = \frac{k_j}{k_j - k_{j-1}} f(t,0,k_j) - \frac{k_{j-1}}{k_j - k_{j-1}} f(t,0,k_{j-1})$$

Index and Tranche NPV as a function of ETL

Given a set of 24 nodal $f(t, k)$ satisfying the constraints and given an interpolation method (linear or spline for example) we will get the 240 $f(t, k)$ we need in order to compute the npv of the tranche and indices for all maturities. At this point we can calculate the theoretical spread that would set exactly to 0 the npv of the instrument if inserted in the premium leg:

$$\mathsf{R}_{\text{Theoretical}} = \frac{\text{PriceDefaultLeg}}{\text{Annuity}} \tag{41}$$

This theoretical spread is a function of the nodal points f we take in the expected tranched loss surface. If the difference between this theoretical spread (function of f) and the mid market quoted spread is smaller than half the bid-ask spread for all instruments, then we have found a set of $f(t, k)$ satisfying the constraints whilst pricing all instruments within the bid-ask spread.

Index and Tranche NPV as a function of ETL

$$\mathrm{Mispr}_{\mathrm{BidAsk}} = \frac{R_{\mathrm{Theoretical}} - R_{\mathrm{MktMid}}}{\mathrm{BidAsk}_{\mathrm{Spread}}/2} \tag{42}$$

Our objective function will be the minimization of the sum of the squared standardized mispricings (42). If using the sum of the squared mispricings will yield a solution for which some of the instruments are priced outside the bid-ask spread ($\mathrm{Mispr}_{\mathrm{BidAsk}} > 1$) then we will try minimizing the sum of even powers of the standardized mispricing (42) for exponents $4, 6$ and 8. In case of persistence of instruments priced outside the bid-ask spread, we take the solution for which the maximum absolute standardized mispricing is smallest.

Index and Tranche NPV as a function of ETL: Numerical Results

	CDX		**ITRAXX**	
sample	from 13-nov-03	to 14-jun-06	from 21-jun-04	to 23-may-06
interpolation	linear	spline	linear	spline
Number of dates	616	616	473	473
% Mispr$_{\textbf{BidAsk}}$ > 1	1.0%	2.6%	0.2%	1.3%
% Mispr$_{\textbf{BidAsk}}$ > 1.2	0.8%	0.2%	0.2%	0.2%
% Mispr$_{\textbf{BidAsk}}$ > 1.4	0.0%	0.0%	0.2%	0.0%
% Mispr$_{\textbf{BidAsk}}$ > 1.6	0.0%	0.0%	0.0%	0.0%

Table 23: Percentage of sample repriced outside the bid-ask range

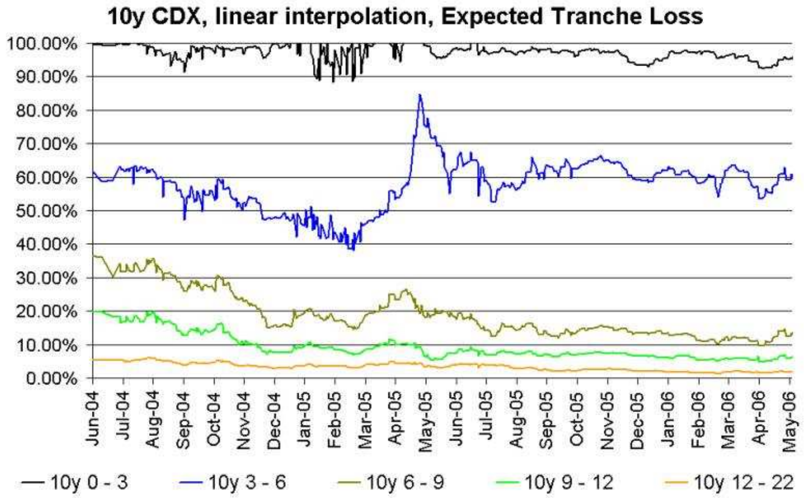
Index and Tranche NPV as a function of ETL: Numerical Results

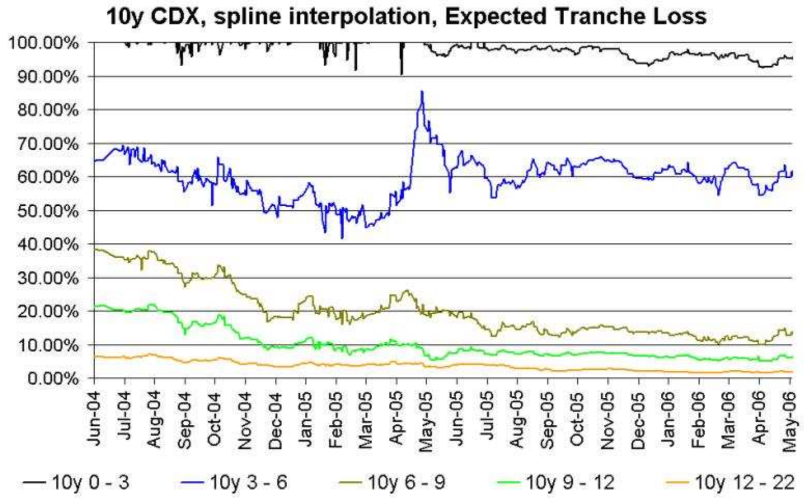
Our sample goes from 13-nov-03 to 14-jun-06 for the CDX and from 21-jun-04 to 23-May-06 for iTraxx. From Table (23) we note that, except for the iTraxx pool with a linear interpolation, in all other cases we find a solution where the theoretical spread exceeds the bid ask spread by less than one fifth $(0.4/2)$ the bid ask range. In the case of the iTraxx pool with a linear interpolation we find only one date where all instruments cannot be priced within the bid ask range: in this case the theoretical spread is outside the bid ask spread by **less** than one third $(0.6/2)$ the bid ask range.

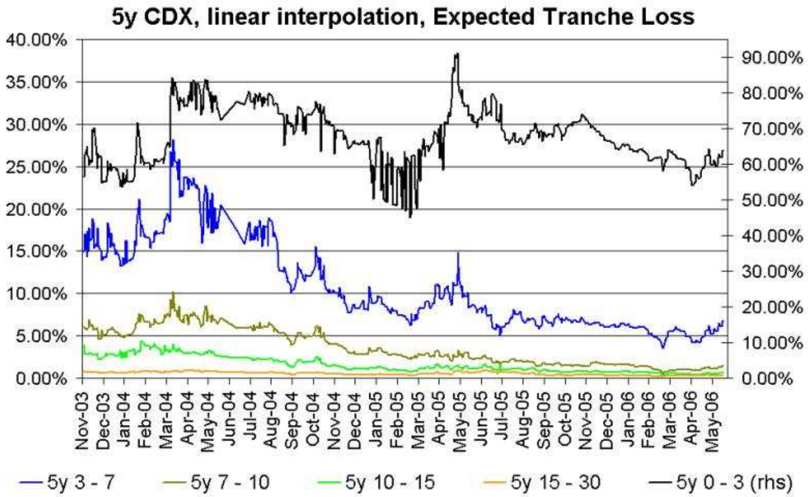
Index and Tranche NPV as a function of ETL: Numerical Results

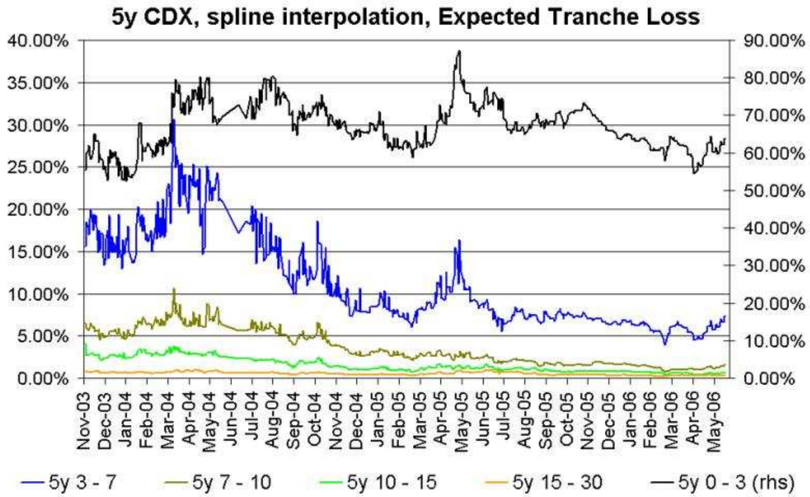
In Figure 30 are plotted the CDX and iTraxx ETL's for the 5y and 10y tranches. It can be clearly noticed the higher perceived riskiness of the CDX universe: despite the higher attachments the ETL is on average higher.

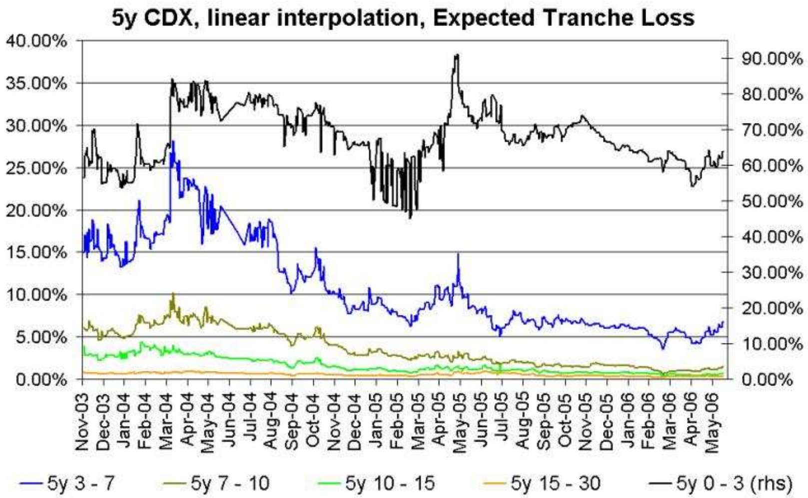
For some dates in our samples we had no quoted market spreads for particular maturities. More specifically, the 3y and 7y tranches are available only from 20-may-05 and 6-may-05 for the CDX and iTraxx respectively. In this cases the unknowns are reduced to be the ETL's on 5 and 10 years maturities only. From Figure 30 we note that were we had only the 5y and 10y tranches available the calibrated 5y and 10y ETL's were much more volatile.

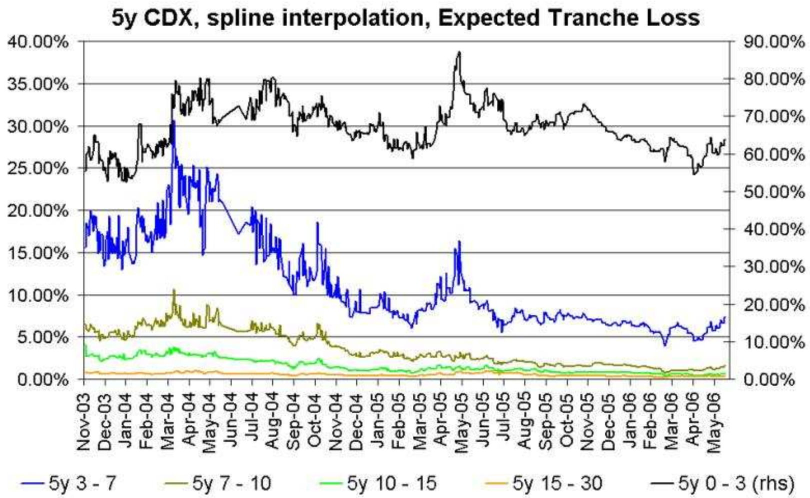

10y CDX, linear interpolation, Expected Tranche Loss
10y 0 - 3
10y 3 - 6
10y 6 - 9
10y 9 - 12
10y 12 - 22

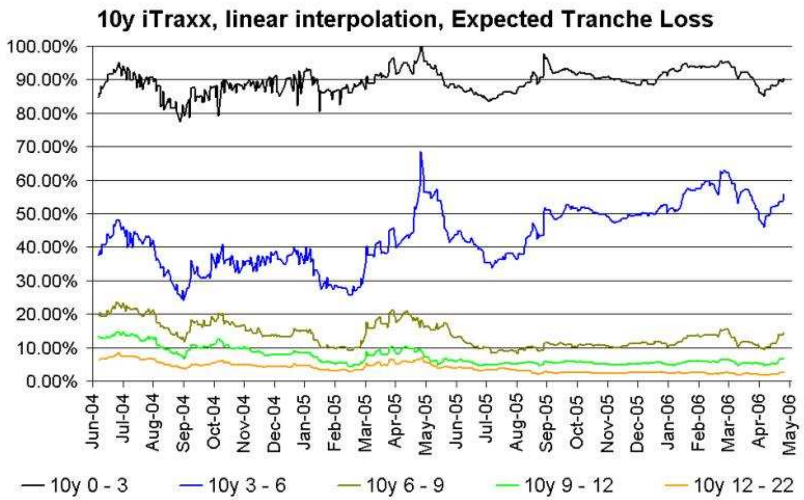

10y CDX, spline interpolation, Expected Tranche Loss
10y 0 - 3
10y 3 - 6
10y 6 - 9
10y 9 - 12
10y 12 - 22

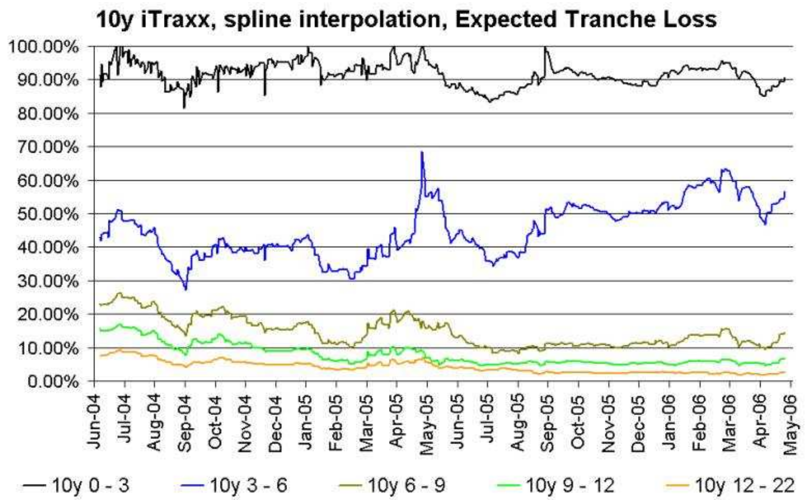

5y CDX, linear interpolation, Expected Tranche Loss
5y 3 - 7
5y 7 - 10
5y 10 - 15
5y 15 - 30
5y 0 - 3 (rhs)


5y CDX, spline interpolation, Expected Tranche Loss
5y 3 - 7
5y 7 - 10
5y 10 - 15
5y 15 - 30
5y 0 - 3 (rhs)


5y CDX, linear interpolation, Expected Tranche Loss
40.00% 35.00% 30.00% 25.00% 20.00% 15.00% 10.00% 5.00% 0.00%
90.00% 80.00% 70.00% 60.00% 50.00% 40.00% 30.00% 20.00% 10.00% 0.00%
Nov-03 Dec-03 Jan-04 Feb-04 Mar-04 Apr-04 May-04 Jun-04 Jul-04 Aug-04 Sep-04 Oct-04 Nov-04 Dec-04 Jan-05 Feb-05 Mar-05 Apr-05 May-05 Jun-05 Jul-05 Aug-05 Sep-05 Oct-05 Nov-05 Dec-05 Jan-06 Feb-06 Mar-06 Apr-06 May-06
5y 3 - 7
5y 7 - 10
5y 10 - 15
5y 15 - 30
5y 0 - 3 (rhs)


5y CDX, spline interpolation, Expected Tranche Loss
40.00% 35.00% 30.00% 25.00% 20.00% 15.00% 10.00% 5.00% 0.00%
90.00% 80.00% 70.00% 60.00% 50.00% 40.00% 30.00% 20.00% 10.00% 0.00%
Nov-03 Dec-03 Jan-04 Feb-04 Mar-04 Apr-04 May-04 Jun-04 Jul-04 Aug-04 Sep-04 Oct-04 Nov-04 Dec-04 Jan-05 Feb-05 Mar-05 Apr-05 May-05 Jun-05 Jul-05 Aug-05 Sep-05 Oct-05 Nov-05 Dec-05 Jan-06 Feb-06 Mar-06 Apr-06 May-06
5y 3 - 7
5y 7 - 10
5y 10 - 15
5y 15 - 30
5y 0 - 3 (rhs)


10y iTraxx, linear interpolation, Expected Tranche Loss
100.00% 90.00% 80.00% 70.00% 60.00% 50.00% 40.00% 30.00% 20.00% 10.00% 0.00%
Jun-04 Jul-04 Aug-04 Sep-04 Oct-04 Nov-04 Dec-04 Jan-05 Feb-05 Mar-05 Apr-05 May-05 Jun-05 Jul-05 Aug-05 Sep-05 Oct-05 Nov-05 Dec-05 Jan-06 Feb-06 Mar-06 Apr-06 May-06
10y 0 - 3
10y 3 - 6
10y 6 - 9
10y 9 - 12
10y 12 - 22


10y iTraxx, spline interpolation, Expected Tranche Loss
100.00% 90.00% 80.00% 70.00% 60.00% 50.00% 40.00% 30.00% 20.00% 10.00% 0.00%
Jun-04 Jul-04 Aug-04 Sep-04 Oct-04 Nov-04 Dec-04 Jan-05 Feb-05 Mar-05 Apr-05 May-05 Jun-05 Jul-05 Aug-05 Sep-05 Oct-05 Nov-05 Dec-05 Jan-06 Feb-06 Mar-06 Apr-06 May-06
10y 0 - 3
10y 3 - 6
10y 6 - 9
10y 9 - 12
10y 12 - 22

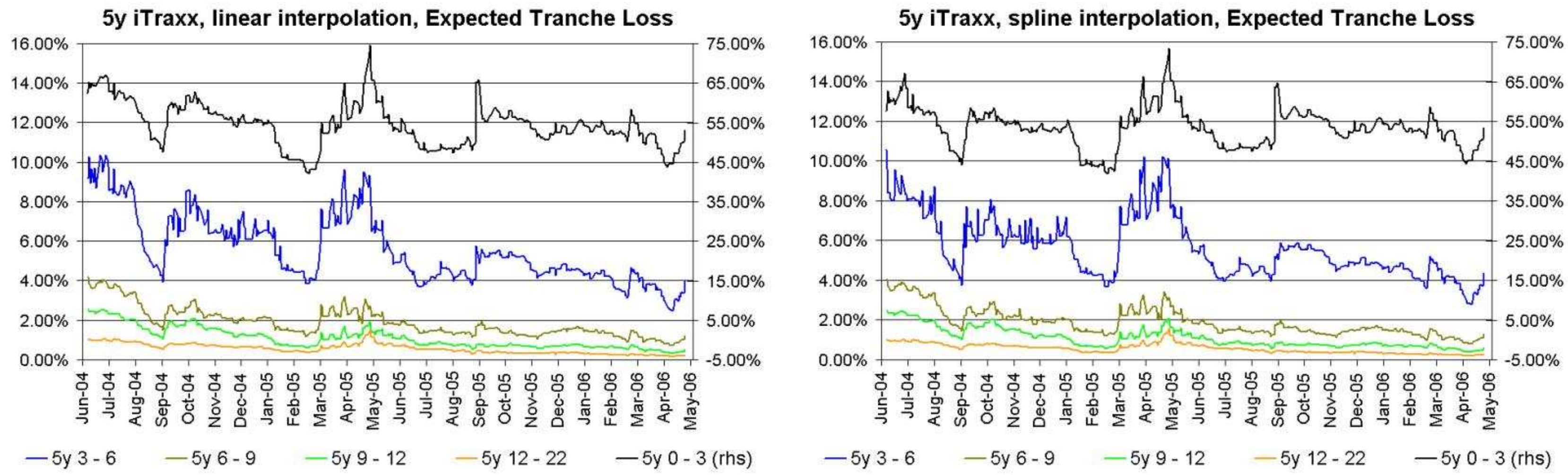


Figure 30: Expected Tranched Loss, CDX and iTraxx, 5y and 10y tranches

Index and Tranche NPV as a function of ETL: Numerical Results

Our results can be reconciled with Walker (2006)'s earlier work.

While in our framework the bid/asks of the instruments enter the target function we aim at minimizing, in Walker (2006)'s framework the instruments npv's must be exactly zero: npv's are treated as a set of linear equality constraints.

Including bid/asks the no-arbitrage constraints are satisfied across the vast majority of dates, we find less violations of the no-arbitrage condition than in Walker's (2006).

Index and Tranche NPV as a function of ETL: Numerical Results

The method appears to be quite powerful as a first *model-independent* procedure to deduce implied expected loss surfaces from market data, allowing one to check basic no-arbitrage constraints in the market quotes.

To price non-standard tranches with attachments or maturities within the observed market range (see for example a $4\% - 10\%$ tranche with $4y$ maturity) we can easily resort to the implied surface in a model independent way.

Index and Tranche NPV as a function of ETL: Numerical Results

We overcome the inconsistency of having different expected loss profiles on the same intervals that would be typically occurring if implied correlation had been used. In Figure 31 we see for example the ETL for the $(0, 3\%)$ equity tranche associated with the $3y$, $5y$ and $10y$ implied correlations for that tranche. We plot the time evolution of Expected tranche losses for $(0, 3\%)$ resulting from implied correlation calibrated to the CDX equity tranches on April 26, 2006. Implied correlation is obtained through inversion of the homogeneos finite pool gaussian copula model price.

As one can see the $[0, 3y]$ expected loss coming from the $5y$ implied correlation quote is different from the $[0, 3y]$ expected loss coming from the 3y correlation quote, and so on. The model independent ETL surface method avoids this inconsistency practically by construction.

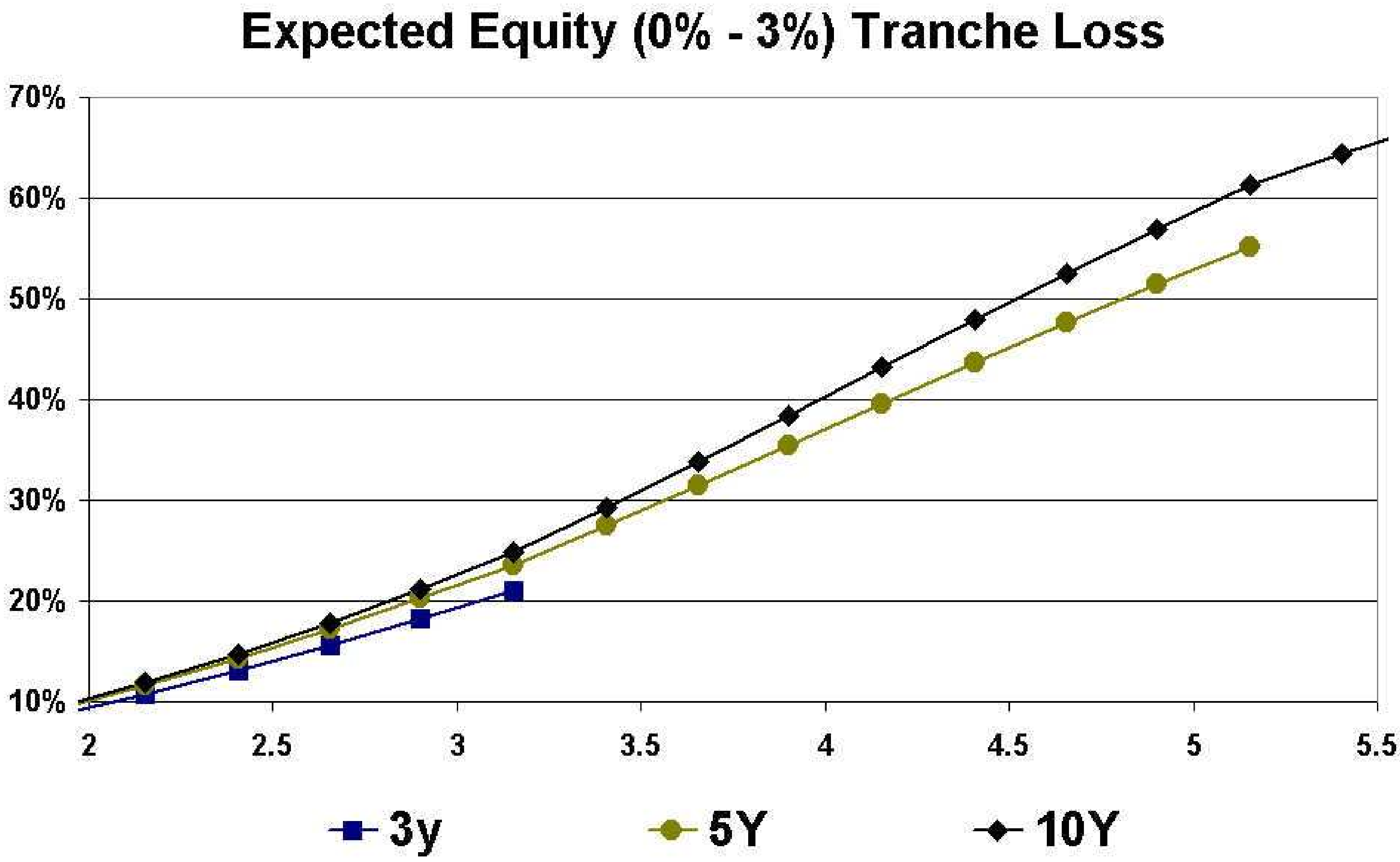
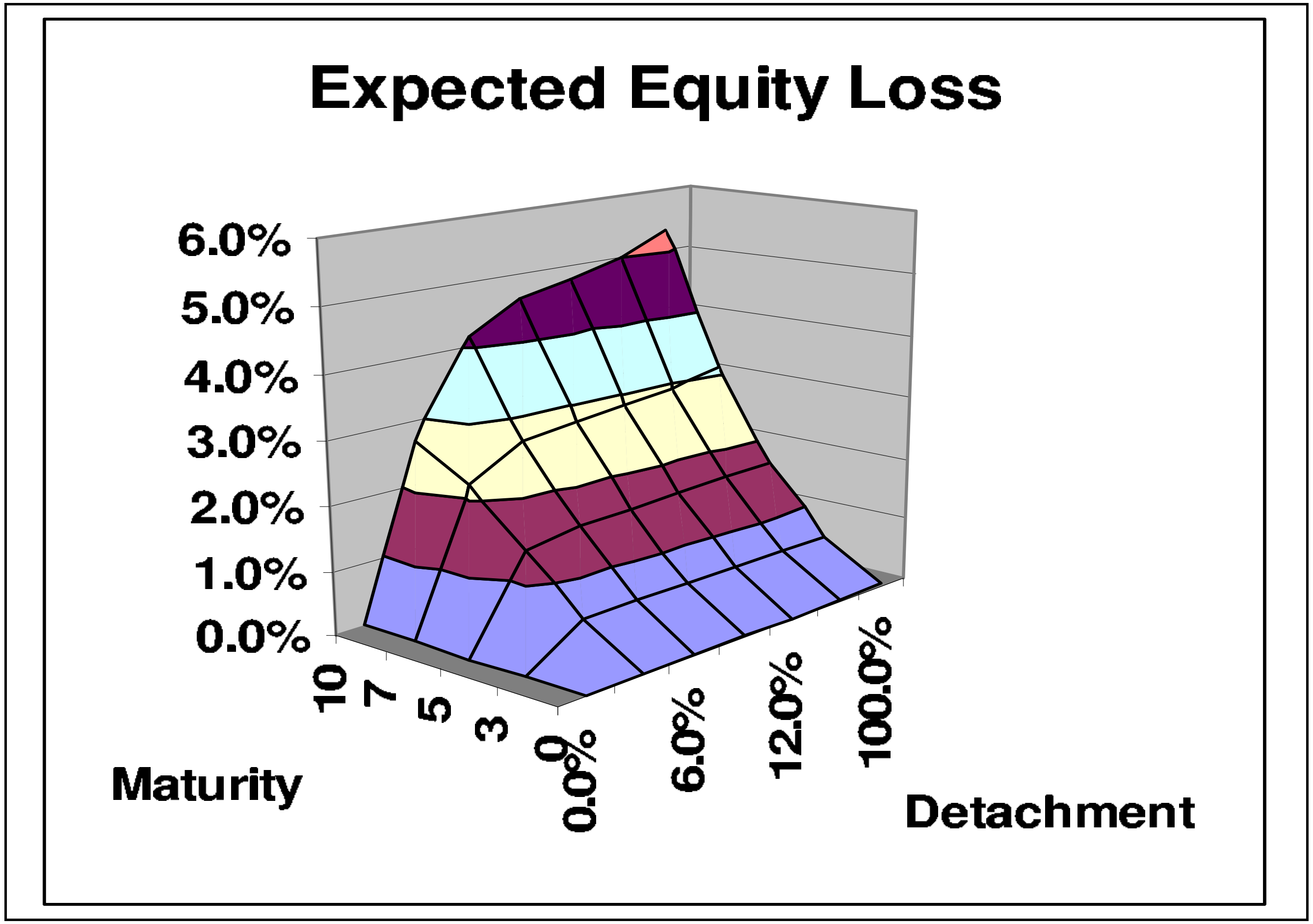


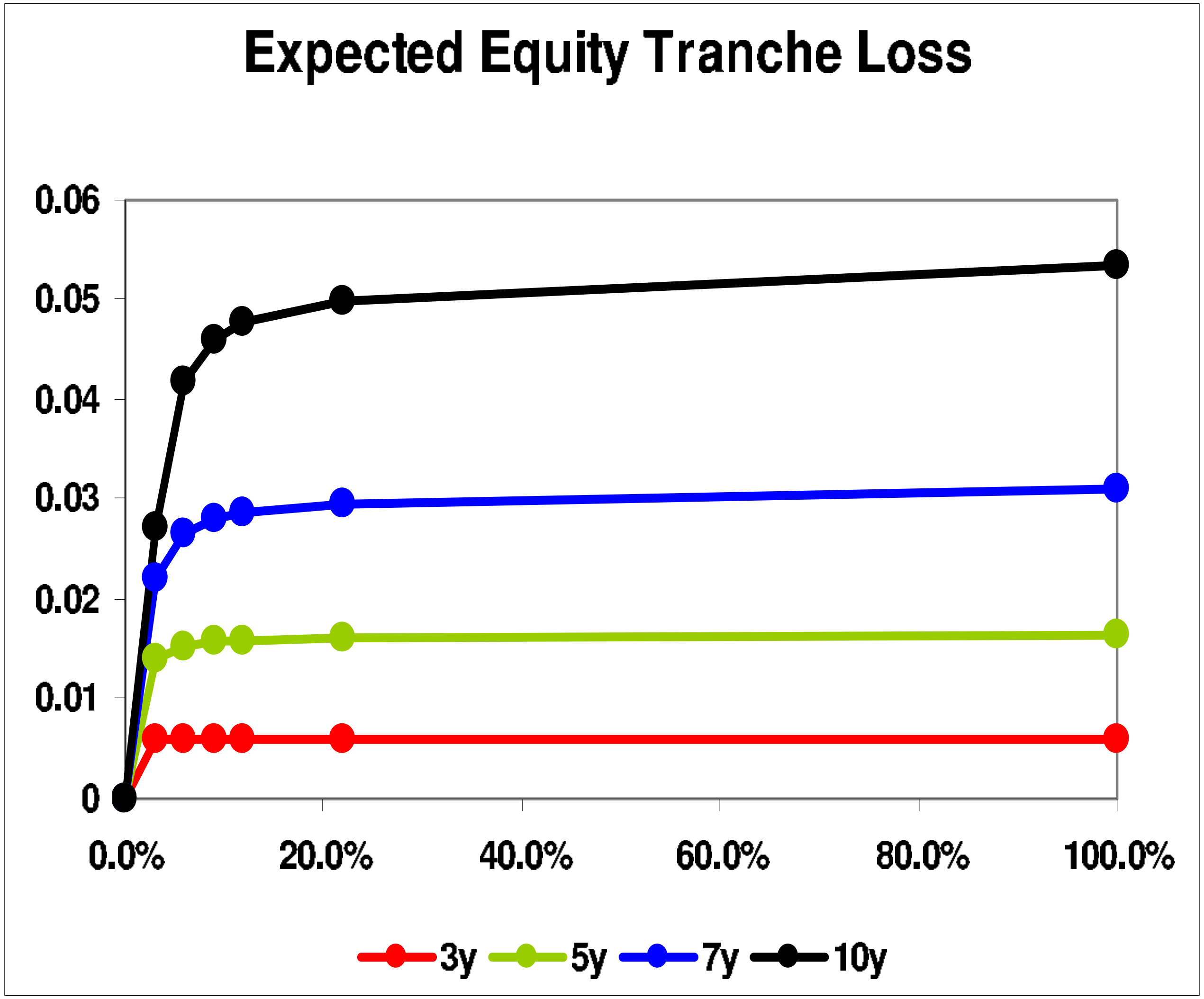
Figure 31: Inconsistency in Expected Tranched Losses coming from implied correlation.

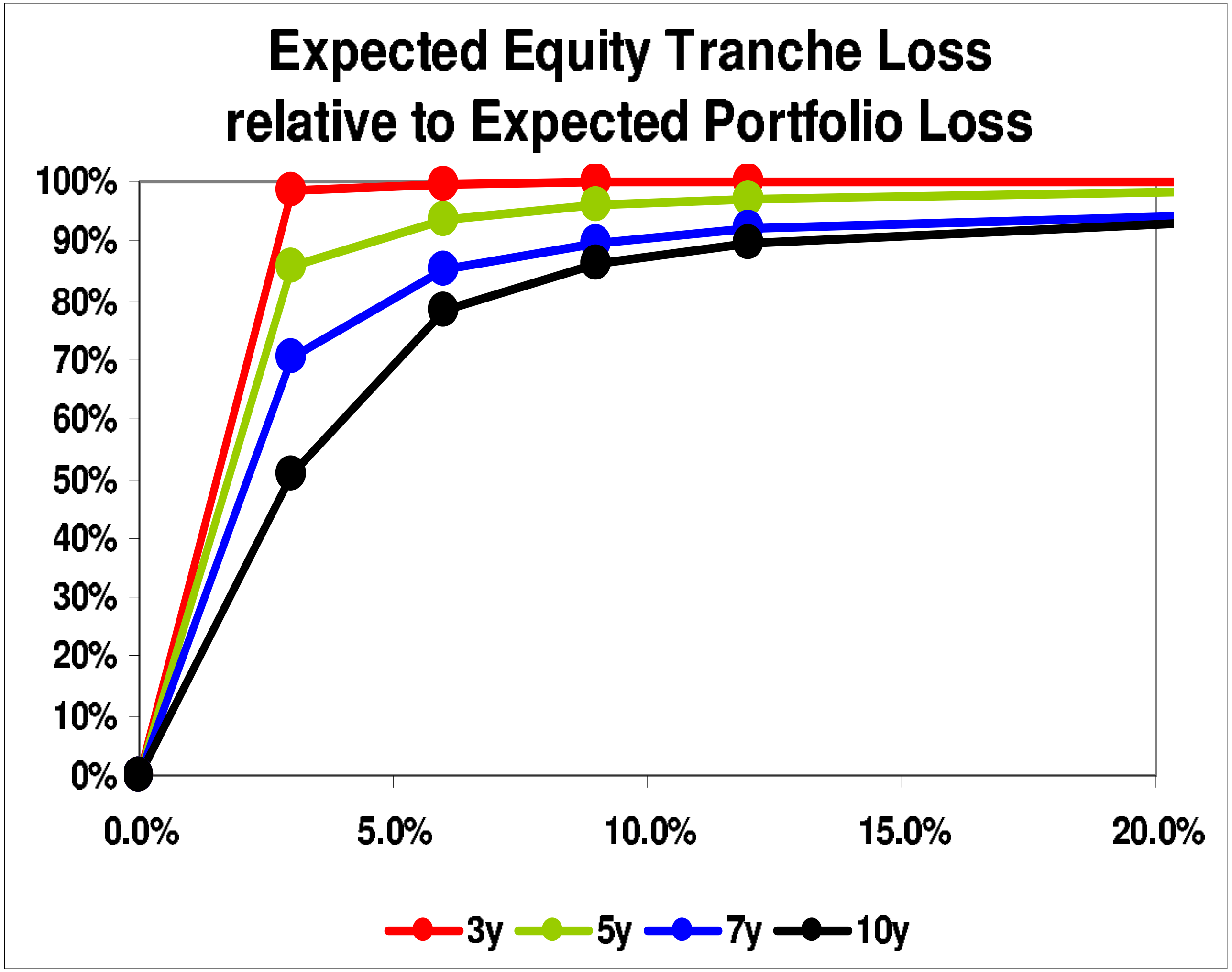
ETL surface: An example from October 2006

	3Y	5Y	7Y	10Y
0-3	2.75%	14.13%	30.38%	45.00%
3-6	4.5	63.5	152	395
6-9	0.75	14.25	43.75	109
9-12	N/A	6.13	19.75	45.75
12-22	N/A	2.5	6.5	14.13
22-100	N/A	0.6	1.7	2.8

Table 24: An example with i-traxx tranche spreads from October 26, 2006


Expected Equity Loss
6.0%
5.0%
4.0%
3.0%
2.0%
1.0%
0.0%
10
7
5
3
0
Maturity
0.0%
6.0%
12.0%
100.0%
Detachment


Expected Equity Tranche Loss
0.06
0.05
0.04
0.03
0.02
0.01
0
0.0%
20.0%
40.0%
60.0%
80.0%
100.0%
3y
5y
7y
10y


Expected Equity Tranche Loss
relative to Expected Portfolio Loss
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0.0%
5.0%
10.0%
15.0%
20.0%
3y
5y
7y
10y

Towards Dynamics Loss Models: Information in CDO quotes

Recall again the market quoted fair spreads for indices and tranches:

$$R_0 = \frac{\mathbb{E}_0\Big[\int_0^T D(0,u)d\textsf{Loss}_u\Big]}{\mathbb{E}_0\Big[\sum_{i=1}^b \delta_i D(0,T_i)(1-\bar{C}_{T_i})\Big]}$$

$$R_0^{A,B} = \frac{\mathbb{E}_0\Big[\int_0^T D(0,u)d\textsf{Loss}_u^{A,B}\Big] - U_0^{A,B}}{\mathbb{E}_0\Big[\sum_{i=1}^b \delta_i D(0,T_i)(1-\textsf{Loss}_{T_i}^{A,B})\Big]}$$

where $\textsf{Loss}_{T_i}^{A,B}$ is the tranched loss at points A, B divided by the tranche thickness $B-A$. C_T is the number of defaults up to T, whereas $\bar{C}_T$ is the same quantity divided by the pool size (default fraction of the pool). Typically $\textsf{Loss}_T = (1-\textsf{Rec})\bar{C}_T$.

If R_0 and $R_0^{A,B}$ are the only data on default correlation in the market, **we see that the only information are "expected losses", "expected tranche losses" and "expected number of defaults".**

Loss models: The “BOTTOM UP” and “TOP DOWN” approaches

Index and tranches contain information only on expected losses, expected tranche losses and expected number of defaults.

Modeling loss and default number? 2 approaches: BOTTOM UP and TOP DOWN.

BOTTOM UP: Model single defaults, correlate them and build the loss from these through recovery assumptions on single names.

TOP DOWN: Model the loss and number of defaults directly as the fundamental objects, and possibly achieve consistency with single names a posteriori.

Loss models: The "BOTTOM UP" approach

BOTTOM UP. We have met this before. In reduced form models, transforming the default time τ by its (strictly increasing) cumulated intensity Λ leads to :

$$\Lambda(\tau) = \xi \sim \text{exponential}, \quad \text{independent of FX, Interest rates, etc..}$$

If we have names $1, 2, ..., n$ we may induce "correlation" among the defaults

$$\tau_1 = \Lambda_1^{-1}(\xi_1), \ldots, \tau_n = \Lambda_n^{-1}(\xi_n)$$

by **putting dependence among the different ξ through a copula.** If one adds recoveries $\text{R}\textsc{ec}_j$, one builds the pool loss from single name losses:

$$\text{L}\textsc{oss}_t = \frac{1}{M}\sum_{j=1}^{M}(1 - \text{R}\textsc{ec}_j)1_{\{\tau_j \le t\}}, \quad \bar{C}_t = \frac{1}{M}\sum_{j=1}^{M} 1_{\{\tau_j \le t\}}$$

Loss models: The "BOTTOM UP" approach

A particular case, with a Gaussian copula collapsing 125 x 124/2= 7750 parameters into 1 is the market "implied correlation" approach.

- **BOTTOM UP:** Easy consistency with single names;
- allows for pricing of CDO squared and other credit payoffs depending on more than the loss of the basic pool; **BUT...**
- The dependence (copula) among single defaults is partly arbitrary;
- Consistent calibration across attachments and maturities is difficult, practically impossible;
- Very difficult to make these models (based on the static notion of copula function) dynamic in order to price forward start CDO or tranche options.

Loss models: The "TOP DOWN" approach

TOP DOWN APPROACH: Model loss-related quantities directly rather than patching single defaults models through a copula.

- a "Market Model" appeal: Focuses on more direct market objects, avoiding arbitrary assumptions on single name default dependencies;
- Possibility to have an authentically dynamic model;
- Calibrate indices and tranches consistently across attachments/maturities;
- Possibility to infer synthetic recovery information on a pool; **BUT...**
- How do losses of different pools "talk" to each other? (CDO squared);
- Consistency with single names: Random Thinning?

Dynamical dependence models: Top Down Approach

Especially in the CDO payoffs, the real underlying of the credit market is the LOSS of the pool rather than single defaults.

Earlier we modeled single default probabilities with intensity models and patched single default together with a copula, building the loss afterwards.

This is called a bottom up approach.

As an alternative, we have then focused on **expected tranche losses (ETL)** as model independent quantities but did not postulate any dynamics underlying them.

Dynamical dependence models: Top Down Approach

Now we go one step further. Rather than contenting ourselves with the ETL, we focus on the Loss as a process to be modeled directly and realistically, and whose dynamics has to be made consistent with market index and tranche data for a start.

This will also allow us to express a more general framework for recovery rates dynamics.

Models that consider directly the aggregate loss and worry later (but not always, unfortunately) about consistency with single name default data are called TOP DOWN.

Dynamical dependence models: Top Down Approach

Recently, some TOP DOWN models for the loss distribution dynamics and/or loss rates have been proposed. These models would be suited to price tranche options, forward starting CDO's and other loss dynamics dependent payoffs.

Some of these models are only "TOP", in that one never goes down to single name level. We will see examples of both TOP (GPL model) and rigorously "TOP DOWN" (GPCL model) approaches.

If the model remains only "TOP", it can be OK when pricing payoffs that are direct functions of the loss, avoiding the cumbersome modeling of the whole detailed dependence structure of default times, but may lead to problems when facing for example CDO squared payoffs or credit payoffs depending on the fine structure of defaults. Further, partial hedges and sensitivities with respect to single names are difficult without the "DOWN" feature.

Top Down Approach: The GPL Model

The basic Generalized Poisson Loss (GPL) model is an example of the TOP (down?) approach and can be formulated as follows.

Consider a number n of independent Poisson processes $N_1, \ldots, N_n$ with intensities $\lambda_1, \ldots, \lambda_n$. Define the stochastic process

$$Z_t = \sum_{j=1}^{n} \alpha_j N_j(t),$$

for increasing integers $\alpha_1, \ldots, \alpha_n$, and model the number of defaults as Z_t.

Top Down Approach: The GPL Model

$$\text{Example :} \quad M = 125, \quad Z_t = 1\ N_1(t) + 2\ N_2(t) + \ldots + 125\ N_{125}(t).$$

If N_1 jumps there has been just one default (idiosyncratic default), if N_{125} jumps there are 125 defaults and the whole pool defaults one shot (systemic risk), otherwise for other N_i's we have intermediate situations.

Some N's may have zero intensity, which is equivalent to say that the corresponding multiplier is set to zero.

This model explicitly contemplates the possibility of multiple defaults in small time intervals, contrary for example to Schönbucher (2005) and Errais, Giesecke and Goldberg (2006).

Top Down Approach: The GPL Model

A drawback of the model is that the number of defaults in time may increase without limit. If our pool contains M names, we may then consider

$$C_t := \min(Z_t, M) = Z_t 1_{\{Z_t < M\}} + M 1_{\{L_t \geq M\}}$$

as actual number of defaults. If Z has a known distribution, the distribution of C_t can be easily derived as a byproduct:

$$\mathbb{Q}(C_t \leq x) = 1_{\{x < M\}} \mathbb{Q}(Z_t < x) + 1_{\{x \geq M\}}$$

The GPL Model

The law of Z_t (and thus of C_t) is directly known through its characteristic function. We have easily, thanks to independence of N_i's,

$$\varphi_{Z_t}(u) = \prod_{j=1}^{n} \mathbb{E}_0[\,\exp(-iu\alpha_j N_j(t))\,] = \prod_{j=1}^{n} \varphi_{N_j(t)}(\alpha_j u),$$

where $\varphi_{N_j(t)}$ is the characteristic function of the Poisson process N_j. Since we know the Poisson char function, we obtain easily

$$\varphi_{Z_t}(u) = \prod_{j=1}^{n} \exp\left[\left(\int_0^t \lambda_j(v)dv\right)\left(e^{i\alpha_j u} - 1\right)\right] = \exp\left[\sum_{j=1}^{n} \Lambda_j(t)\left(e^{i\alpha_j u} - 1\right)\right]$$

The density of Z_t can be obtained as the inverse Fourier transform

The GPL Model

The process Z_t can also be characterized as a so called generalized Poisson process (GPP): a Poisson process with the possibility to allow for multiple jumps.

A GPP J_t is defined as a process with stationary independent increments, and the increments may amount to integers $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_n$. The probability to have a jump of size α_k given that there has been at least one jump of any positive size satisfies

$$\lim_{h \to 0} \mathbb{Q}\{J_{t+h} - J_t = \alpha_k | J_{t+h} - J_t \geq \alpha_1\}/h = p_k$$

(would be zero in Poisson). Also, the probability of having no jumps up to time t and to have at least one jump in arbitrarily small times is

$$\mathbb{Q}(J_t = 0) = \exp(-\lambda t), \ \lim_{h \to 0} \mathbb{Q}\{J_h > 0\}/h = \lambda$$

exactly as for the Poisson process.

The GPL Model

Now let us go back to Z_t. In case we take time homogeneous Poisson processes N_j with constant intensities λ_j, our process above for Z_t is the same as a GPP J_t with the same α's and multiple jump probabilities p_i and intensity λ given by

$$p_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}, \qquad \lambda = \sum_{j=1}^{n} \lambda_j .$$

In other terms, our linear combination of Poisson processes is the same as a generalized Poisson process allowing for multiple jumps with given probabilities.

One more way of looking at our process is the compound Poisson process, see Brigo, Pallavicini and Torresetti (2006) for the details on this interpretation. The counting process is Poisson with intensity $\lambda_1 + \ldots + \lambda_n$ and jumps are i.i.d. discrete r.v.'s each taking value α_i with probability $\lambda_i / \sum_{j=1}^{n} \lambda_j$.

Default intensity

An important feature of loss models is to link default intensities jumps to loss dynamics, so that the default intensity decreases, as long as loss increases, and it is equal to zero when the whole portfolio has defaulted.

Let us consider the compensator A_t of the default-counting point process C_t, namely the nondecreasing predictable process that added to a local martingale gives C_t itself (Doob-Meyer decomposition), satisfying

$$\mathbb{E}_t[\, C_T - A_T \,] = C_t - A_t.$$

Default intensity

A can be computed as

$$A_T := \lim_{h\downarrow 0}\int_0^T \frac{\mathbb{E}_t[\,C_{t+h} - C_t\,]}{h}dt = \lim_{h\downarrow 0}\int_0^T \frac{\mathbb{E}_t\big[\,\min(Z_{t+h} - Z_t, M - Z_t)1_{\{Z_t<M\}}\,\big]}{h}dt$$

$$= \lim_{h\downarrow 0}\int_0^T \frac{\mathbb{E}_t\Big[\,\sum_{j=1}^n \min(\alpha_j, M - Z_t)1_{\{Z_t<M\}}1_{\{Z_{t+h}-Z_t=\alpha_j\}}\,\Big]}{h}dt$$

$$= \lim_{h\downarrow 0}\int_0^T \frac{\sum_{j=1}^n \min(\alpha_j, M - Z_t)1_{\{Z_t<M\}}\mathbb{E}_t\Big[\,1_{\{Z_{t+h}-Z_t=\alpha_j\}}\,\Big]}{h}dt$$

Default intensity

so that, with a final calculation,

$$A_T = \int_0^T \sum_{j=1}^{n} \min(\alpha_j, (M - Z_{t^-})^+) \lambda_j(t) dt \tag{43}$$

where we have taken the left limit in the integrand to ensure its left-continuity (and hence predictability). If A_t is absolutely continuous, its density is known as the intensity of the process C_t, and is given by

$$h_C(t) = \sum_{j=1}^{n} \min(\alpha_j, (M - Z_{t^-})^+) \lambda_j(t). \tag{44}$$

The default intensity h_C in the basic GPL model is a stochastic object only through Z_t. It is possible to extend the GPL model by considering the intensities λ_j as stochastic processes, e.g. following a Gamma or CIR process. The default intensity h_C acquires a new source of stochasticity.

The Gamma GPL Model

One interesting extension is the Gamma-intensity Generalized Poisson (GGPL) model.

$$Z_t^{\mathrm{G}} = \sum_{j=1}^{n} \alpha_j N_j^{\mathrm{G}}(t),$$

where now $N_j^{\mathrm{G}}(t)$ are Cox processes (i.e. Poisson processes with stochastic intensity) whose random cumulated intensities are distributed at any time T as

$$\int_0^T \lambda_j(t)dt =: \Lambda_j(T) \sim \Gamma(k_j(T), \theta_j)$$

where $k > 0$ is the shape parameter and $\theta > 0$ is the scale parameter in the Gamma distribution. We take different $\Lambda_j(T)$ to be independent as j changes.

The Gamma GPL Model

We still compute the characteristic function in closed form as

$$\begin{aligned}\varphi^{\mathrm{G}}_{Z_T}(u) &= \mathbb{E}_0\Big[\,\mathbb{E}_0\Big[\,\exp(iuZ^{\mathrm{G}}_T)\;\big|\;\Lambda_1(T),\ldots,\Lambda_n(T)\Big]\,\Big] = \\ &= \prod_{j=1}^{n}\left[\left(1+\theta_j\left(1-e^{i\alpha_j u}\right)\right)\right]^{-k_j(T)}\end{aligned}$$

The Gamma distribution assumption $\Lambda_j(t) \sim \Gamma(k_j(t), \theta_j)$ at every time is consistent with a gamma process assumption for Λ_j, for more details and a piecewise Gamma extension allowing for tractability and a term structure in the parameter θ see Brigo, Pallavicini and Torresetti (2006), where we further discuss the case of stochastic scenario intensities.

The CIR-GPL Model

A different and possibly more interesting extension is the CIR- Generalized Poisson (CIR-GPL) model

$$Z_t^{\mathrm{CIR}} = \sum_{j=1}^{n} \alpha_j N_j^{\mathrm{CIR}}(t), \qquad d\lambda_j = k_j(\theta_j - \lambda_j)dt + \sigma_j \sqrt{\lambda_j} dW_j,$$

with $2k_j\theta_j > \sigma_j^2$, and where the intensities of multiple defaults with different sizes follow different independent CIR processes.

The characteristic function of Z can be computed again in closed form, the calculation being quite similar to the bond price formula for the CIR interest rate model. Alternatively, **jump diffusion** JCIR intensities can be considered, maintaining tractability.

Extended GPL Models: Spread Dynamics

In general the stochastic intensity may help us to model **volatility** when considering for example **forward start CDO's or tranche options**.

For example, consider the index future spread at t:

$$R_t = \frac{\int_t^T D(t,u)\mathbb{E}_t[\, h_L(u)\,]\, du}{\sum_{i=1}^n 1_{\{T_i > t\}} \delta_i D(t,T_i)\,(1 - \bar{C}_t - \int_t^{T_i} \mathbb{E}_t[\, h_{\bar{C}}(u)\,]\, du)}$$

and recall the compensator,

$$h_C(t) = \sum_{j=1}^n \min(\alpha_j, (M - Z_t)^+)\lambda_j(t)$$

and similarly for h_{LOSS}. CIR intensities λ_j for the spreads may enrich the dynamics of the index spread.

Recovery assumptions

In order to ensure an arbitrage-free dynamics, the portfolio cumulated loss (Loss_t) and the re-scaled number of defaults ($\bar{C}_t$) must be non-decreasing processes taking values in the $[0, 1]$ interval, the former with increments always smaller or equal than the increment of the latter.

$$d\mathsf{Loss}_t \leq d\bar{C}_t.$$

When we write expression like dX_t for jump processes X (that we assume to be right continuous with left limit) we mean $X_t - X_{t-}$, where X_{t-} is the left limit of X at t.

Recovery assumptions

The portfolio cumulated loss and the number of defaults cannot be independetly modelled, since they are coupled by the forward realization of the recovery rate (REC_t) at default

$$d\text{LOSS}_t = [1 - \text{REC}_t] d\bar{C}_t,$$

where we define REC_t as the "recovery rate at default", assuming it is a $\mathcal{G}_t$-adapted and left continuous (and hence predictable) process taking values in the interval $[0, 1]$.

Recovery assumptions

The recovery rate can be expressed also in terms of the intensities of the loss and default rate processes. By taking the expectation on both sides of

$$d\text{Loss}_t = [1 - \text{Rec}_t]d\bar{C}_t, \tag{45}$$

conditional on $\mathcal{G}_{t-}$, we obtain

$$\text{Rec}_t = 1 - \frac{h_{\text{Loss}}(t)}{h_{\bar{C}}(t)} \tag{46}$$

Equation (63) shows that in general the recovery rate at default is directly related to the intensities of both the loss and the default rate processes. Thus, the choice for the intensity dynamics does induce a dynamics for the recovery rate.

As a first approach we choose to model $\bar{C}$ as in our GPL approach, to introduce a constant recovery rate $\text{Rec} = 40\%$ and to let Equation (45) define the Loss.

Calibration

The GPL model is calibrated to the market quotes observed on March 1 and 6, 2006. Deterministic discount rates are listed in Brigo, Pallavicini and Torresetti (2006). Tranche data and DJi-TRAXX fixings, along with bid-ask spreads, are

	Att-Det	**March, 1 2006**		**March, 6 2006**		
		5y	7y	3y	5y	7y
Index		35(1)	48(1)	20(1)	35(1)	48(1)
Tranche	0-3	2600(50)	4788(50)	500(20)	2655(25)	4825(25)
	3-6	71.00(2.00)	210.00(5.00)	7.50(2.50)	67.50(1.00)	225.50(2.50)
	6-9	22.00(2.00)	49.00(2.00)	1.25(0.75)	22.00(1.00)	51.00(1.00)
	9-12	10.00(2.00)	29.00(2.00)	0.50(0.25)	10.50(1.00)	28.50(1.00)
	12-22	4.25(1.00)	11.00(1.00)	0.15(0.05)	4.50(0.50)	10.25(0.50)
Tranchlet	0-1	6100(200)	7400(300)			
	1-2	1085(70)	5025(300)			
	2-3	393(45)	850(60)			

Calibration

The cumulated intensities $\Lambda_i(T)$ are real non-decreasing piecewise linear functions in the tranche maturity.

The optimal values for the amplitudes α are selected as follows:

1. set $\alpha_1 = 1$ and all other α's to zero. Calibrate Λ_1;
2. find the best integer value for α_2 by calibrating the cumulated intensities Λ_1 and Λ_2 for each value of α_2 in the range $[1, 125]$,starting from the previous Λ_1 as a guess;
3. repeat the previous step for α_i with $i = 3$ and so on, by calibrating the cumulated intensities $\Lambda_1, \ldots, \Lambda_i$, starting from the previously found $\Lambda_1, \ldots, \Lambda_{i-1}$ as initial guess, until the calibration error is under a pre-fixed threshold or until the intensity Λ_i can be considered negligible.

Calibration

The objective function f to be minimized in the calibration is the squared sum of the errors shown by the model to recover the tranche and index market quotes weighted by market bid-ask spreads:

$$f(\alpha, \Lambda) = \sum_i \epsilon_i^2, \quad \epsilon_i = \frac{x_i(\alpha, \Lambda) - x_i^{\mathrm{Mid}}}{x_i^{\mathrm{Bid}} - x_i^{\mathrm{Ask}}}$$

where the x_i, with i running over the market quote set, are the index values R_0 for DJi-TRAXX index quotes, and either the index periodic premium rates $R_0^{A,B}$ or the upfront premium rates $U_0^{A,B}$ for the DJi-TRAXX tranche quotes.

Calibration: All standard tranches up to seven years

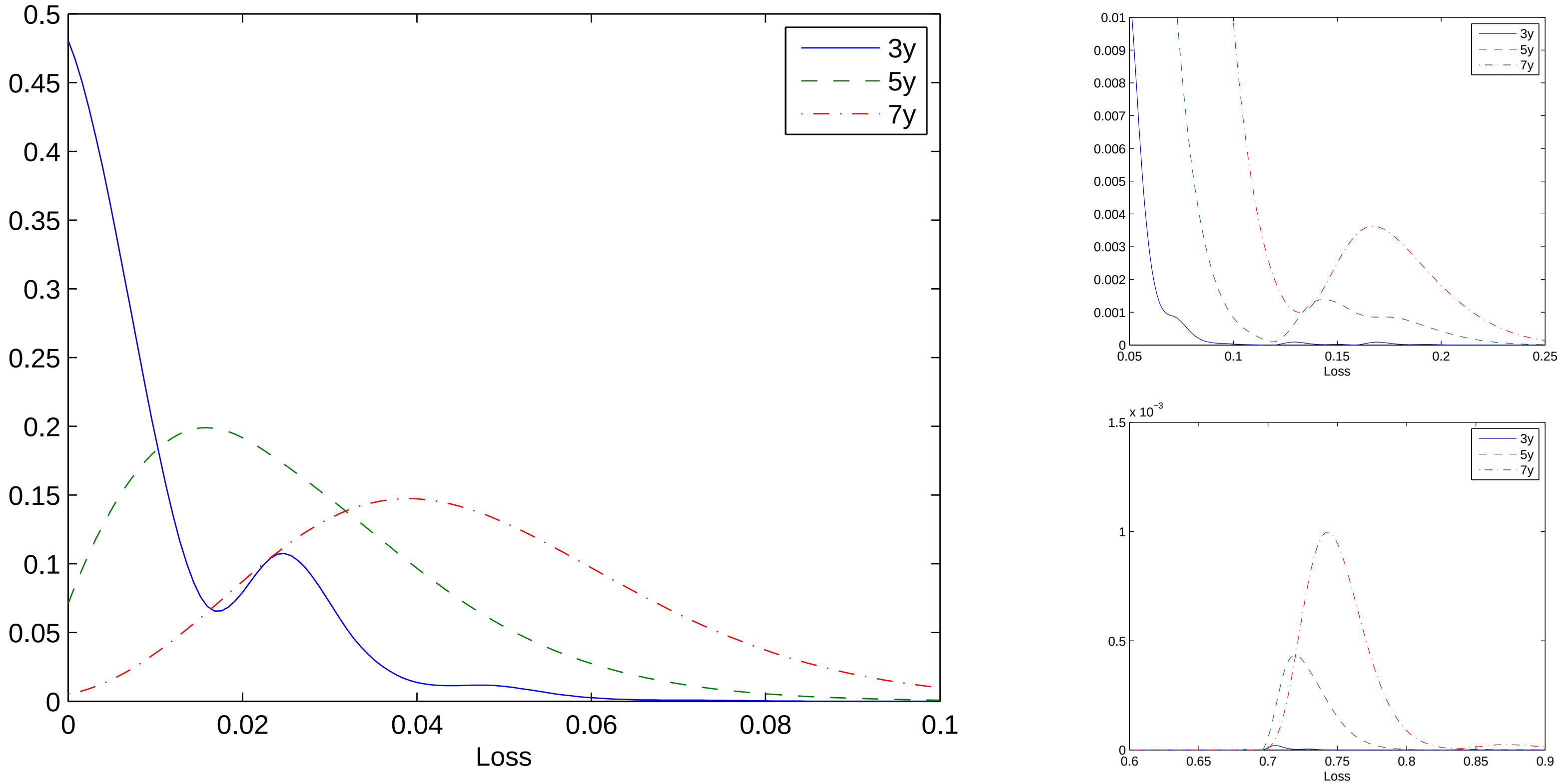
As a first calibration example we consider standard DJi-TRAXX tranches up to a maturity of 7y with constant recovery rate of 40%.

The calibration procedure selects five Poisson processes. The 18 market quotes used by the calibration procedure are almost perfectly recovered. In particular all instruments are calibrated within the bid-ask spread (we show the ratio calibration error / bid ask spread).

	Att-Det	**Maturities**		
		3y	5y	7y
Index		-0.4	-0.2	-0.9
Tranche	0-3	0.1	0.0	-0.7
	3-6	0.0	0.0	0.7
	6-9	0.0	0.0	-0.2
	9-12	0.0	0.0	0.0
	12-22	0.0	0.0	0.2

$\boldsymbol{\alpha}$	$\boldsymbol{\Lambda(T)}$		
	3y	5y	7y
1	0.535	2.366	4.930
3	0.197	0.266	0.267
16	0.000	0.007	0.024
21	0.000	0.003	0.003
88	0.000	0.002	0.007

Calibration: All standard tranches up to seven years



Calibration: All standard tranches up to seven years

One possible comparison of our implied loss distribution according to the GPL model is with the implied loss distribution according to the (static) "implied copula" approach seen earlier.

If we compare the implied loss distribution resulting from the calibration of the five year index and tranche quotes with the implied copula approach as reformulated in Torresetti et al. (2006) (Implied Default Rate approach, seen earlier in Feature article 1), we find a qualitative pattern similar to the pattern we have above.

Calibration: All standard tranches up to seven years

Notice in particular the large portion of mass concentrated near the origin, the subsequent modes when moving along the loss distribution for increasing values, and the bumps in the far tail.

These features are common to both approaches. In our GPL models the bumps in the tails of the loss distributions, which seem to be necessary in order to be able to recover the market quotes, are obtained thanks to the multiple jumps components contributing to the loss distribution. In particular, the components with higher α's are giving rise to the little bumps in the far tail of the loss distribution and help with senior tranches.

Calibration: Tranchlets

The market quotes also non-standard tranches, which are quoted over the counter. An interesting case is given by the so called "tranchlets", namely DJi-TRAXX tranches with attachment and detachment points possibly smaller than 3%. On the first of march 2006 we obtain market quotes for a set of tranchlets with maturity of five and seven years (see earlier table).

We calibrate the market data with constant recovery rate of 40%. The calibration procedure selects five Poisson processes. The 18 market quotes used by the calibration procedure are recovered, but within an error that is occasionally larger than the bid-ask spread.

Calibration: Tranchlets

	Att-Det	**Maturities**	
		5y	7y
Index		-0.8	-2.1
Tranchlet	0-1	1.1	-1.4
	1-2	1.7	-0.6
	2-3	-0.1	-0.4
Tranche	0-3	0.1	0.4
	3-6	-1.9	0.2
	6-9	0.4	0.6
	9-12	2.8	0.9
	12-22	-0.4	-1.5

$\boldsymbol{\alpha}$	$\boldsymbol{\Lambda(T)}$	
	5y	7y
1	0.834	3.336
2	1.070	1.070
13	0.008	0.015
21	0.004	0.013
104	0.002	0.007

Table 25: Left side: calibration error with respect to the bid-ask spread for tranches quoted by the market. Right side: cumulated intensities of the basic GPL model. Each row corresponds to a different Poisson component with jump amplitude α. Recovery rate is 40%.

Calibration: More recent results (2006). i-Traxx

	Att-Det	Maturities			
		3y	5y	7y	10y
Index		18(0.5)	30(0.5)	40(0.5)	51(0.5)
Tranche	0-3	350(150)	1975(25)	3712(25)	4975(25)
	3-6	5.50(4.0)	75.00(1.0)	189.00(2.0)	474.00(4.0)
	6-9	2.25(3.0)	22.25(1.0)	54.25(1.5)	125.50(3.0)
	9-12		10.50(1.0)	26.75(1.5)	56.50(2.0)
	12-22		4.00(0.5)	9.00(1.0)	19.50(1.0)
	22-100		1.50(0.5)	2.85(0.5)	3.95(0.5)

Table 26: DJi-TRAXX index and tranche quotes in basis points on October 2, 2006, along with the bid-ask spreads. Index and tranches are quoted through the periodic premium, whereas the equity tranche is quoted as an upfront premium.

$\boldsymbol{\alpha_j}$	$\boldsymbol{\Lambda_j(T)}$			
	3y	5y	7y	10y
1	0.778	1.318	3.320	4.261
3	0.128	0.536	0.581	1.566
15	0.000	0.004	0.024	0.024
19	0.000	0.007	0.011	0.028
32	0.000	0.000	0.000	0.007
79	0.000	0.000	0.003	0.003
120	0.000	0.002	0.003	0.008

Table 27: DJi-TRAXX pool. Cumulated intensities, integrated up to tranche maturities, of the basic GPL model. Each row j corresponds to a different Poisson component with jump amplitude α_j. The recovery rate is 40%. All tranches are calibrated within one bid-ask

Calibration: More recent results (2006). CDX

	Att-Det	Maturities			
		3y	5y	7y	10y
Index		24(0.5)	40(0.5)	49(0.5)	61(0.5)
Tranche	0-3	975(200)	3050(100)	4563(200)	5500(100)
	3-7	7.90(1.6)	102.00(6.1)	240.00(48.0)	535.00(21.4)
	7-10	1.20(0.2)	22.50(1.4)	53.00(10.6)	123.00(7.4)
	10-15	0.50(0.1)	10.25(0.6)	23.00(4.6)	59.00(3.5)
	15-30	0.20(0.1)	5.00(0.3)	7.20(1.4)	15.50(0.9)

Table 28: CDX index and tranche quotes in basis points on October 2, 2006, along with the bid-ask spreads. Index and tranches are quoted through the periodic premium, whereas the equity tranche is quoted as an upfront premium.

$\boldsymbol{\alpha_j}$	$\boldsymbol{\Lambda_j(T)}$			
	3y	5y	7y	10y
1	1.132	3.043	4.247	7.166
2	0.189	0.189	0.812	1.625
6	0.011	0.091	0.091	0.091
18	0.000	0.006	0.028	0.028
23	0.000	0.004	0.005	0.032
32	0.000	0.000	0.000	0.009
124	0.000	0.003	0.005	0.010

Table 29: CDX pool. Left side: cumulated intensities, integrated up to tranche maturities, of the basic GPL model. Each row j corresponds to a different Poisson component with jump amplitude α_j. The recovery rate is 40%. All tranches are calibrated within one bid-ask

GPL: Self-exciting features

$$\text{Recall} \quad h_C(t) = \sum_{j=1}^{n} \min(\alpha_j, (M - Z_t)^+)\lambda_j(t)$$

and similarly for $h_{\text{LOSS}} = h_C(1 - \text{REC})/M$.

If all the possible integer jump sizes between 1 and M are allowed, i.e. if $\alpha_j = j$ and $n = M$, the intensity h_{LOSS} jumps whenever the cumulated portfolio loss process LOSS jumps. The intensity jumps downwards, and this would seem to go in the opposite direction with respect to self-excitedness, which is considered a desirable feature of loss models in general.

However, self-exciting features are embedded in our model, and they are embedded in the possibility to have several defaults in small intervals, contrary to most approaches to loss modeling. Consider for example just two names: instead of having the loss of one name increase the likelihood of default (intensity) of a second name, we have both names defaulting together immediately. This embeds self-excitedness, although in an extreme way.

Pricing and Extensions

Pricing products based on the loss distribution such as tranche options, forward start tranches etc with the calibrated model is simple, given knowledge of the marginal and transition distributions for the constituent Poisson processes.

Indeed, if we have a payoff or additive portion of a payoff depending on the loss at one maturity, we simply sample one-shot the independent Poissons N_j at maturity, add them up using the related multiplicity coefficients α, plug the resulting loss in the payoff portion and average over scenarios.

This is maintained also under random (Gamma, scenario or CIR) intensities. Alternatively, we may decide to use the inverse Fourier transform of the known characteristic function of the terminal distribution to obtain the loss density and then integrate numerically the payoff against this density.

Pricing and Further research

If a payoff is path dependent on the loss we still may simulate the independent increments of the independent constituent Poisson processes N_j among the relevant instants.

Given independence this can be realized by sampling known independent Poisson laws. Once this has been done, we obtain the constituent processes at every relevant time by adding up their increments, and then we obtain the loss at any time by simply adding the constituent processes times their multiplicity coefficients α.

Then we plug each temporal path of the loss distribution in the payoff and average over scenarios. This procedure is substantially maintained also under possible random (Gamma, scenario or CIR) intensities. Simulation is thus easy and based on the ability to sample from a Poisson law.

Single names? Products depending on the fine default structure?

For products such as CDO squared, depending not only on the aggregate loss of the pool but also on losses of sub-pools, one needs more than the aggregate loss dynamics.

Also, for hedge ratios with respect to single name spreads and default probabilities, one needs to make the model consistent with possibly available single name default data (CDS etc.).

How can one do this?

A general method that is oftern advocated is “random thinning”. This is based on “slicing” the pool compensator h_C in therms of single name defaults compensators. See for example Giesecke and Goldberg.

However, random thinning is doubtful in practice and except under peculiar assumptions on the dynamics of the pool and single name intensities, it may be difficult to apply.

Single names? Products depending on the fine default structure?

Alternatively, Lindskog and McNeil (2003) with the Common Poisson Shock (CPS) framework and Elouerkhaoui (2006) find that a model similar to ours is consistent with a bottom up model with Marshall Olkin copula across single names. This also could achieve consistency with single defaults.

The problem with CPS is that defaults are not limited. A name may default more than once (actually infinite times) and the number of names that may default in time in a finite pool is also infinite.

We try and adjust the CPS framework, avoiding repeated defaults and finding a framework that in the aggregate looks similar to GPL: the Generalized Poisson Cluster-adjusted Loss model (GPCL).

GPCL: The Common Poisson Shocks Framework

We begin by briefly illustrating the common Poisson shock framework (CPS), reviewed for example in Lindskog and McNeil (2003).

The occurrence of a default in a pool of names can be originated by different events, either idiosyncratic or systematic. In the CPS framework, the occurrence of the event number e, with $e = 1 \ldots m$, is modelled as a jump of a Poisson process $N^{(e)}$. Notice that each event can be triggered many times. Poisson processes driving different events are considered to be independent.

The CPS setup assumes unrealistically that a defaulted name k may default again. Later we try and limit the number of defaults of each name to one.

GPCL: The Common Poisson Shocks Framework

For now, we assume that the r-th jump of $N^{(e)}$ triggers a default event for the name k with probability $p^{(e)}_{r,k}$, leading to the following dynamics for the single name default process N_k, defined as the process that jumps each time name k defaults:

$$N_k(t) := \sum_{e=1}^{m} \sum_{r=1}^{N^{(e)}(t)} I^{(e)}_{r,k}$$

where $I^{(e)}_{r,k}$ is a Bernoulli variable with probability $\mathbb{Q}\{I^{(e)}_{r,k} = 1\} = p^{(e)}_{r,k}$. Under the Poisson assumption for N^e and the Bernoulli assumption for $I^{(e)}_{r,j}$ it follows that N_k is itself a Poisson process. Notice however that the processes N_k and N_h followed by two different names k and h are not independent since their dynamics is explained by the same driving events.

GPCL: The Common Poisson Shocks Framework

The core of the CPS framework consists in mapping the single name default dynamics, consisting of the dependent Poisson processes N_k, into a multi-name dynamics explained in terms of *independent* Poisson processes $\widetilde{N}_s$, where s is a subset (or "cluster") of names of the pool, defined as follows.

$$\widetilde{N}_s(t) = \sum_{e=1}^{m} \sum_{r=1}^{N^{(e)}(t)} \sum_{s' \supseteq s} (-1)^{|s'|-|s|} \prod_{k' \in s'} I_{r,k'}^{(e)}$$

where $|s|$ is the number of names in the cluster s. In a summation, $s \ni k$ means we are adding up across all clusters s containing k, $k \in s$ means we are adding across all elements k of cluster s, while $|s| = j$ means we are adding across all clusters of size j and, finally, $s' \supseteq s$ means we are adding up across all clusters s' containing cluster s as a subset.

GPCL: The Common Poisson Shocks Framework

The non-trivial proof of the independence of $\widetilde{N}_s$ for different subsets s can be found in Lindskog and McNeil (2003). Notice that a jump in a $\widetilde{N}_s$ processes means that all the names in the subset s, *and only those names*, have defaulted at the jump time. We denote by $\widetilde{\lambda}_s$ the intensity of the Poisson process $\widetilde{N}_s(t)$, and we assume it to be deterministic for the time being, although we present extensions later.

Cluster processes $(\widetilde{N}_s)$ as CPS building blocks

One does not need to remember the above construction. All that matters are the independent clusters default Poisson processes $\widetilde{N}_s(t)$. These can be taken as fundamental variables from which (correlated) single name defaults and default counting processes follow. The single name dynamics can be derived based on these independent $\widetilde{N}_s$ processes:

$$N_k(t) = \sum_{s \ni k} \widetilde{N}_s(t), \quad \text{or} \quad dN_k(t) = \sum_{s \ni k} d\widetilde{N}_s(t), \tag{47}$$

where the second equation is the same as the first one but in instantaneous jump form. We now introduce the process $Z_j(t)$, describing the occurrence of the simultaneous default of any j names whenever it jumps (with jump-size one):

$$Z_j(t) := \sum_{|s|=j} \widetilde{N}_s(t). \tag{48}$$

Notice that each $Z_j(t)$, being the sum of independent Poisson processes, is itself Poisson. Further, since the clusters corresponding to the different $Z_1, Z_2, \ldots, Z_M$ never overlap, the $Z_j(t)$ are independent Poisson processes.

Cluster processes $(\widetilde{N}_s)$ as CPS building blocks

The multi-name dynamics, that is the default counting process Z_t for the whole pool, can be easily derived by carefully adding up all the single name contributions.

$$Z_t := \sum_{k=1}^{M} N_k(t) = \sum_{k=1}^{M} \sum_{s \ni k} \widetilde{N}_s(t) = \sum_{k=1}^{M} \sum_{j=1}^{M} \sum_{s \ni k, |s|=j} \widetilde{N}_s(t) = \sum_{j=1}^{M} j \sum_{|s|=j} \widetilde{N}_s(t),$$

leading to the relationship which links the set of dependent single name default processes N_k with the set of independent and Poisson distributed counting processes Z_j:

$$\sum_{k=1}^{M} N_k(t) = \sum_{j=1}^{M} j Z_j(t) =: Z_t \quad (49)$$

Cluster processes $(\widetilde{N}_s)$ as CPS building blocks

Hence, the CPS framework offers us a way to consistently model the single name processes along with the pool counting process taking into account the correlation structure of the pool, which remains specified within the definition of each cluster process $\widetilde{N}_s$. Notice, however, that the Z_t/M process is not properly the re-scaled number of defaults $\bar{C}_t$, since the former can increase without limit, while the latter is bounded in the $[0, 1]$ interval. We address this issue, a reflection of the problem that in CPS single names may default more than once, later.

CPS: Copula structure of default times

The single name default dynamics in the CPS framework induces a Marshall-Olkin copula type dependence between the first jumps of the single name processes N_j. More precisely, if the random default times $\{\tau_1, \ldots, \tau_M\}$ of names $1, \ldots, M$ in the pool are modeled as the first jump times of the single name processes $N_1, \ldots, N_M$,

$$\tau_k := \inf\{t \geq 0 : N_k(t) > 0\},$$

then Lindskog and McNeil (2003) show that the default times vector is distributed according to a multi-variate distribution whose survival copula is a M-dimensional Marshall-Olkin copula.

Improving CPS: Avoiding repeated defaults

In the above framework we have a fundamental problem, due to repeated jumps of the same Poisson processes. Indeed, if the jumps are to be intepreted as defaults, this leads the above framework to unrealistic consequences. Repeated defaults would occur both at the cluster level, as $\widetilde{N}_s$ keeps on jumping, and at the single name level, since each Poisson process N_k keeps on jumping. These repetitions would cause the default counting process Z_t to exceed the pool size M and to grow without limit in time.

There are two main strategies to solve this problem. Both take as starting points the cluster repeated-default processes $\widetilde{N}_s$ and then focus on different variables. They can be summarized as follows.

Improving CPS: Avoiding repeated defaults

Strategy 1 (Single-name adjusted approach). Force single name defaults to jump only once and deduce clusters jumps consistently.

Strategy 2 (Cluster adjusted approach). Force clusters to jump only once and deduce single names defaults consistently.

The two choices have different implications, and we explore both of them in the following. The second solution is more promising.

If one gives up single names and clusters, and focuses only on the default counting process and the loss (throwing away the “bottom-up” interpretation), there is a third possible strategy to make the default counting process above consistent with the pool size:

Strategy 0 (Default-counting adjusted approach, GPL). Modify the aggregated pool default counting process so that this does not exceed the number of names.

Improving CPS: Avoiding repeated defaults

Strategy 0, i.e. our earlier GPL model, addresses the problem of the CPS framework at the default counting level. In the basic CPS framework, the link between the re-scaled pool counting process Z_t/M, which can increase without limit, and the re-scaled number of defaults $\bar{C}_t$, that must be bounded in the $[0,1]$ interval, is not correct. This forbids in principle to model $\bar{C}_t$ as Z_t/M. In the CPS literature this problem is not considered usually. Lindskog and McNeil (2003) for instance suppose that the default intensities of the names are so small to lead to negligible "second-default" probabilities. If this assumption were realistic, this would allow for adopting Z_t/M as a model for $\bar{C}_t$ and strategy 0 would not be needed. However, in our calibration results of GPL above, we find that intensities are large enough to make repeated defaults unacceptable in practice.

Improving CPS: Avoiding repeated defaults

In Table 30 we summarize the notation we are going to adopt in the following.

	default process for cluster s	default proc for name k	default count proc for j simult defaults	total default counting
Repeated defaults	$\widetilde{N}_s$	N_k	Z_j	Z
Strategy 0 (GPL)	–	–	Z_j^0	$\min(Z, M)$
Strategy 1	$\widetilde{N}_s^1$	N_k^1	Z_j^1	Z^1
Strategy 2 (GPCL)	$\widetilde{N}_s^2$	N_k^2	Z_j^2	Z^2

Table 30: Notation for default processes according to the different strategies

Single-name adjusted approach (strategy 1)

In order to avoid repeated defaults in single name dynamics, we can introduce constraints on the single name dynamics ensuring that each single name makes only one default. Given the same repeated cluster processes $\widetilde{N}_s$ as before, we *define* the new single name default processes N_k^1 replacing N_k as solutions of

$$\begin{aligned} dN_k^1(t) &:= (1 - N_k^1(t^-)) \sum_{s \ni k} d\widetilde{N}_s(t) \qquad (50) \\ &= \sum_{s \ni k} d\widetilde{N}_s(t) \prod_{s \ni k} 1_{\{\widetilde{N}_S(t^-)=0\}} \end{aligned}$$

Interpretation: *This equation amounts to say that name k jumps at a given time if some cluster s containing k jumps (i.e. $\widetilde{N}_s$ jumps) and if no cluster containing name k has ever jumped in the past.*

Single-name adjusted approach (strategy 1)

We can compute the new cluster defaults $\widetilde{N}_s^1$ consistent with the single names N_k^1 as

$$d\widetilde{N}_s^1(t) = \prod_{j\in s} dN_j^1(t) \prod_{j\in s^C} (1 - dN_j^1(t)) \tag{51}$$

where s^c is the set of all names that do not belong in s.

Single-name adjusted approach (strategy 1)

Now, we calculate how the new counting processes Z_j^1 are to be defined in terms of the new single names default dynamics:

$$\begin{aligned}\sum_{k=1}^{M} dN_k^1(t) &= \sum_{k=1}^{M}(1-N_k^1(t^-))\sum_{s\ni k} d\widetilde{N}_s(t) = \sum_{k=1}^{M}(1-N_k^1(t^-))\sum_{j=1}^{M}\sum_{s\ni k,|s|=j} d\widetilde{N}_s(t)\\ &= \sum_{j=1}^{M}\sum_{|s|=j} d\widetilde{N}_s(t)\sum_{k\in s}(1-N_k^1(t^-)) = \sum_{j=1}^{M}\sum_{|s|=j} d\widetilde{N}_s(t)\sum_{k\in s}\prod_{s'\ni k} 1_{\{\widetilde{N}_{s'}(t^-)=0\}}.\end{aligned}$$

This expression should match $dZ^1(t) := \sum_j j\ dZ_j^1(t)$, so that the counting processes are to be defined as

$$dZ_j^1(t) := \frac{1}{j}\sum_{|s|=j} d\widetilde{N}_s(t)\sum_{k\in s}\prod_{s'\ni k} 1_{\{\widetilde{N}_{s'}(t^-)=0\}} \tag{52}$$

Single-name adjusted approach (strategy 1)

The intensities of the above processes can be directly calculated in terms of the density of the process compensator. We obtain by direct calculation

$$
\begin{aligned}
h_{N_k^1}(t) &= \prod_{s \ni k} 1_{\{\widetilde{N}_s(t^-)=0\}} \sum_{s \ni k} \widetilde{\lambda}_s(t) \\
h_{Z_j^1}(t) &= \frac{1}{j} \sum_{|s|=j} \widetilde{\lambda}_s(t) \sum_{k \in s} \prod_{s' \ni k} 1_{\{\widetilde{N}_{s'}(t^-)=0\}}
\end{aligned}
$$

where in general we denote by $h_X(t)$ the compensator density of process X at time t, referred to as "intensity of X", and where $\widetilde{\lambda}_s$ is the intensity of the Poisson process $\widetilde{N}_{s'}$.

Single-name adjusted approach (strategy 1)

Given exogenously the repeated Poisson "cluster" default building blocks $\widetilde{N}_s$, the model $N_k^1, \widetilde{N}_s^1, Z_j^1$ is a consistent way of simulating the single name processes, the cluster processes and the pool counting process from the point of view of avoiding repeated defaults. In particular, we obtain $\bar{C}_t := \sum_k N_k^1(t)/M = Z_t^1/M \leq 1$.

Notice, however, that the definition of N_k^1, even if it avoids repeated defaults of single names, is not consistent with the spirit of the original repeated cluster dynamics.

Consider indeed the following example.

Single-name adjusted approach (strategy 1)

Begin Example. *Consider two clusters* $s = \{1, 2, 3\}$, $z = \{3, 4, 5, 6\}$*. Assume no name defaulted up to time* t *except for cluster* z*, in that in a single past instant preceding* t *names* $3, 4, 5, 6$ *(and only these names) defaulted together (*$\widetilde{N}_z$ *jumped at some past instant). Now suppose at time* t *cluster* s *jumps, i.e. names* $1, 2, 3$ *(and only these names) default, i.e.* $\widetilde{N}_s$ *jumps for the first time.*

Question: Does name 2 *default at* t*?*

According to our definition of N_2^1 *the answer is yes, since no cluster containing name* 2 *has ever defaulted in the past. However, we have to be careful in interpreting what is happening at* cluster *level. Indeed, clusters* z *and* s *cannot both default since this way name* 3 *(that is in both clusters) would default twice. So we see that the actual clusters default of this approach do not have a clear intuitive link with repeated cluster defaults* $\widetilde{N}_s$*.*

End Example.

Single-name adjusted approach (strategy 1)

To simplify the parameters, we may assume the cluster intensities $\widetilde{\lambda}_s$ to depend only on the cluster size $|s| = j$. Then it is possible to directly calculate the intensity of the pool counting process $C = Z^1$ as

$$h_{Z^1}(t) = \left(1 - \frac{Z^1_{t^-}}{M}\right) \sum_j j \binom{M}{j} \widetilde{\lambda}_j$$

where $\widetilde{\lambda}_j$ is the common intensity of clusters of size j.

Single-name adjusted approach (strategy 1)

We see that the pool counting process intensity h_{Z^1} is a linear function of the counting process $C = Z^1$ itself, as we can expect by general arguments for a pool of *independent* names (again with homogeneous intensities). In such a pool default of one name does not affect the intensity of default of other names, and the pool intensity is the common homogeneous intensity times the number of outstanding names. Each new default simply diminishes the pool intensity of one common intensity value and the pool intensity is always proportional to the number (fraction) of outstanding names $(1 - \bar{C})$.

GPCL model: Cluster-adjusted approach (strategy 2)

The key to *consistently* avoid repeated cluster defaults (and subsequently single names) is to track, when a cluster jumps, which single-name defaults are triggered, and then force all the clusters containing such names not to jump any longer.

GPCL model: Cluster-adjusted approach (strategy 2)

We may formalize these points by introducing the process $J_s(t)$ defined as

$$J_s(t) := \prod_{k \in s} \prod_{s' \ni k} 1_{\{\widetilde{N}_{s'}(t)=0\}} = \prod_{s':\, s' \cap s \neq \emptyset} 1_{\{\widetilde{N}_{s'}(t)=0\}}$$

The process $J_s(t)$ is equal to 1 at starting time and it jumps to 0 whenever a cluster containing one element of s jumps. Or one may view the process J_s as being one when none of the names in s have defaulted and 0 when some names in s have defaulted. Notice that $J_s(t) = 1$ implies $1_{\{\widetilde{N}_s(t)=0\}}$ but not viceversa.

GPCL model: Cluster-adjusted approach (strategy 2)

We now correct the cluster dynamics by avoiding repeated clusters defaults. We define as new cluster dynamics the following:

$$d\widetilde{N}_s^2(t) = J_s(t^-)d\widetilde{N}_s(t). \tag{53}$$

Interpretation: *every time a repeated cluster default process* $\widetilde{N}_s$ *jumps, this is a jump in our "no-repeated-jumps" framework only if no name contained in* s *has defaulted in the past, i.e. if no cluster intersecting* s *has defaulted in the past.*

GPCL model: Cluster-adjusted approach (strategy 2)

Once the clusters defaults are given, single name defaults follow easily. Define the single name dynamics as

$$dN_k^2(t) := \sum_{s \ni k} d\widetilde{N}_s^2 = \sum_{s \ni k} J_s(t^-) d\widetilde{N}_s(t). \tag{54}$$

Now, re-define default counting processes in terms of our new cluster dynamics. We obtain

$$dZ^2{}_j := \sum_{|s|=j} d\widetilde{N}_s^2 = \sum_{|s|=j} J_s(t^-) d\widetilde{N}_s(t). \tag{55}$$

GPCL model: Cluster-adjusted approach (strategy 2)

The pool counting process reads

$$dZ^2 = \sum_{j=1}^{M} j \sum_{|s|=j} d\widetilde{N}_s^2 = \sum_{j=1}^{M} j \sum_{|s|=j} J_s(t^-) d\widetilde{N}_s(t). \quad (56)$$

If not for the cluster-related indicators $J_s(t^-)$, Z^2 would be a generalized Poisson process. That is why we term the model $N_k^2, \widetilde{N}_s^2, {Z^2}_j$ the Generalized Poisson Cluster-adjusted Loss model (GPCL).

GPCL model: Cluster-adjusted approach (strategy 2)

To appreciate how this second strategy formulation improves on the first strategy, we consider again our earlier example.

Example (Reprise). *Consider the same example as before up to the Question: "Does name 2 default at t?"*

According to our definition of N_2^2 the answer is now NO, since the cluster $z = \{3, 4, 5, 6\}$, intersecting the s currently jumping (they both have name 3 as element), has already defaulted in the past. Thus we see a clear difference between strategies 1 and 2. With strategy 2 name 2 does not default when s jumps, with strategy 1 it does. Notice that strategy 2 is more consistent with the original spirit of the repeated cluster defaults $\widetilde{N}_s$. Indeed, if cluster $z = \{3, 4, 5, 6\}$ has defaulted in the past (meaning that $\widetilde{N}_z$ has jumped), $s = \{1, 2, 3\}$ should never be allowed to default, since it is impossible that now "exactly the names $1, 2, 3$ default", given that 3 has already defaulted in z.

End Example

GPCL model: Cluster-adjusted approach (strategy 2)

The intensities of the above processes can be directly calculated as densities of the processes compensators. We obtain by direct calculation, given that $J_s(t)$ is known given the information (and in particular the $\widetilde{N}_s$) at time t:

$$h_{N_k^2}(t) = \sum_{s \ni k} J_s(t^-)\widetilde{\lambda}_s(t) \tag{57}$$

$$h_{Z_j^2}(t) = \sum_{|s|=j} J_s(t^-)\widetilde{\lambda}_s(t) \tag{58}$$

GPCL model: Cluster-adjusted approach (strategy 2)

(Self-affecting features). Notice that in the GPCL model the single name intensities $h_{N_k^2}(t)$ are stochastic, since they depend on the process J_s. Moreover, the single name intensities are affected by the loss process. In particular, the intensity of a single-name jumps when one of the other names jumps. Consider for example a name k that has not defaulted by t, with intensity $h_{N_k^2}(t)$, and one path where there are no new defaults until $t' > t$, when name k' defaults. Now all clusters s containing k' have $J_s(t') = 0$ so that

$$h_{N_k^2}(t') = \sum_{s \ni k} J_s(t'^-)\widetilde{\lambda}_s(t') = \sum_{s \ni k} J_s(t^-)\widetilde{\lambda}_s(t') - \sum_{s \supseteq \{k,k'\}} J_s(t^-)\widetilde{\lambda}_s(t')$$

We see that the the k-th name intensity reduces when k' defaults, and it reduces of the second summation in the last term.

GPCL model: Cluster-adjusted approach (strategy 2)

At first sight this is a behaviour that is not ideally suited to intensities. For example, looking at the loss feedback present in the default intensities of Hawkes-processes (see Errais, Giesecke and Goldberg (2006) for Hawkes processes applied to default modeling), one sees that intensities are self-exciting, in that they *increase* when a default arrives. As soon as one name defaults, the intensities of the pool jump up, as is intuitive. However, Errais, Giesecke and Goldberg (2006) (but also Schönbucher (2005) and others) assume there is only one default at a time. We are instead assuming there may be more than one default in a single instant. Therefore the self-exciting feature is somehow built in the fact that more than one name may default at the same instant. In other terms, instead of having the intensity of default of a related name jumping up of a large amount, implying that the name could default easily in the next instants, we have the two names defaulting together. From this point of view cluster defaults embed the self-exciting feature, although in an extreme way.

GPCL model: Cluster-adjusted approach (strategy 2)

The best way to summarize our construction is through the three equations defining respectively cluster defaults, single name defaults and default counting processes:

$$d\widetilde{N}^2_s(t) = J_s(t^-)d\widetilde{N}_s(t), \quad \boxed{dN^2_k(t) := \sum_{s \ni k} d\widetilde{N}^2_s(t), \; dZ^2_j(t) := \sum_{|s|=j} d\widetilde{N}^2_s(t)}$$

Notice that once the new cluster default processes $\widetilde{N}^2_s$ are properly defined, single name and default counting processes follow immediately in what is indeed the only possible relationships that make sense for connecting clusters fatal shocks to single name defaults and to default counting processes.

GPCL model: Cluster-adjusted approach (strategy 2)

We may also write the cluster intensities as

$$h_{\widetilde{N}_s^2}(t) = J_s(t^-)\widetilde{\lambda}_s(t) =: \bar{\lambda}_s(t)$$

Notice that this strongly reminds us of what we do with Poisson (or more generally Cox) processes to model single name defaults. The default time τ_k of the single name k is modeled as the first jump of a Poisson process with intensity $\lambda_k(t)$, and then the process is killed after the first jump in order to avoid repeated defaults. This way the intensity $\bar{\lambda}_k(t)$ of the default time τ_k is

$$\bar{\lambda}_k(t) = 1_{\{\tau_k > t\}}\lambda_k(t)$$

What we do is similar for clusters: we start from clusters with repeated jumps $\widetilde{N}_s$ and then we kill the repeated jumps through an indicator $J_s(t)$, replacing the simpler indicator $1_{\{\tau_k > t\}}$ of the single-name case.

GPCL model: Cluster-adjusted approach (strategy 2)

If, as before, *we assume the cluster intensities* $\widetilde{\lambda}_s$ *to depend only on the cluster size,* $\widetilde{\lambda}_s = \widetilde{\lambda}_{|s|}$, it is possible to directly calculate the intensity of the pool counting process $Z^2(t) := \sum_j j Z_j^2(t)$. We obtain

$$h_{Z^2}(t) = \sum_j j \binom{M - Z^2_{t^-}}{j} \widetilde{\lambda}_j$$

where $\widetilde{\lambda}_j$ is the common intensity of clusters of size j. The pool counting process intensity is a non-linear function of the counting process, taking into account the co-dependence of single name defaults.

Beyond GPL: The GPCL model calibration

WE have seen earlier the GPL basic model $C_t = \min(Z_t, M)$, calibrated to the index and its tranches for several maturities.

Here we try instead the richer GPCL model $C_t = Z_t^2$ introduced above, allowing us in principle to model also cluster and single name defaults consistently.

However, the GPCL model can hardly be managed without simplifying assumptions. In the following we assume again that the cluster intensities $\widetilde{\lambda}_s$ depend only on the cluster size $|s|$, and focus only on calibration of multiname products.

Differently from GPL, our GPCL default counting process intensity has a clear interpretation in terms of default clusters. This will allow us, in further work, to ecplicitly include single names in the picture, since our GPCL framework allows us to do so explicitly.

Beyond GPL: The GPCL model calibration

The recovery rate is considered as a deterministic constant and set equal to $R = 40\%$. Thus, the underlying driving model definition is

$$C_t := Z^2(t) = \sum_{j=1}^{M} j\, Z_j^2(t), \quad \text{where} \quad dZ_j^2(t) \sim \text{Poisson}\left(\binom{M - Z_{t^-}^2}{j} \widetilde{\lambda}_j(t) dt \right)$$

while the pool counting and loss processes are defined as

$$\begin{aligned} d\bar{C}_t &:= dZ_t^2/M \\ d\mathsf{Loss}_t &:= (1-R)\, dZ_t^2/M \end{aligned}$$

Beyond GPL: The GPCL model calibration

Given our recovery assumption, the prices of the products to be calibrated, depend only on knowledge of the probability distribution of the pool counting process C_t. Thus, our main issue is to calculate this law as fast as possible.

With the GPCL model, the dependence of the intensity of the pool counting process on the process itself prevents us either to calculate the relevant characteristic function in closed form (as for GPL instead) or to use the Panjer method.

Beyond GPL: The GPCL model calibration

Our choice then is to explicitly calculate the forward Kolmogorov equation satisfied by the probability distribution $p_{Z_t^2}(x) = \mathbb{Q}Z_t^2 = x$, namely

$$\frac{d}{dt} p_{Z_t^2}(x) = \sum_{y=0}^{M} A_t(x,y) p_{Z_t^2}(y)$$

where the transition rate matrix $A_t = (A_t(x,y))_{x,y=0,\ldots,M}$ is given by

$$A_t(x,y) := \lim_{\Delta t \to 0} \frac{\mathbb{Q}Z_{t+\Delta t}^2 = x | Z_t^2 = y}{\Delta t} = \binom{M-y}{x-y} \widetilde{\lambda}_{x-y}(t)$$

for $x > y$,

$$A_t(y,y) := \lim_{\Delta t \to 0} \frac{\mathbb{Q}Z_{t+\Delta t}^2 = y | Z_t^2 = y-1}{\Delta t} = -\sum_{j=1}^{M-y} \binom{M-y}{j} \widetilde{\lambda}_j(t).$$

for $x = y$, and zero for $x < y$.

Beyond GPL: The GPCL model calibration

In matrix form we write

$$\frac{d}{dt}\widehat{\pi}_t = A_t \widehat{\pi}_t, \qquad \widehat{\pi}_t := \left[\begin{array}{ccccc} p_{Z_t^2}(0) & p_{Z_t^2}(1) & p_{Z_t^2}(2) & \ldots & p_{Z_t^2}(M) \end{array} \right]'$$

whose solution is obtained through the exponential matrix,

$$\widehat{\pi}_t = \exp\left(\int_0^t A_u du \right) \widehat{\pi}_0, \;\; \widehat{\pi}_0 = [1\ 0\ 0 \ldots\ 0]'.$$

Matrix exponentiation can be quickly computed with the Padé approximation (see Golub and Van Loan (1983)), leading to a closed form solution for the probability distribution $p_{C_t} = \widehat{\pi}_t$ of the pool counting process C_t. This distribution can then be used in the calibration procedure.

Beyond GPL: The GPCL model calibration

If we define the cumulated cluster intensities as

$$\widetilde{\Lambda}_j(t) = \int_0^t \widetilde{\lambda}_j(u)\, du.$$

then the entries of the matrix undergoing exponentiation in determining the default counting distribution are given by

$$\text{for } x > y\text{:}\quad \int_0^t A_u(x,y)du = \binom{M-y}{x-y}\widetilde{\Lambda}_{x-y}(t)$$

$$\text{for } x = y\text{:}\quad \int_0^t A_u(y,y)du = -\sum_{j=1}^{M-y}\binom{M-y}{j}\widetilde{\Lambda}_j(t).$$

We assume the $\widetilde{\Lambda}_j$ to be piecewise linear in time, changing their values at payoff maturity dates. We use $\widetilde{\Lambda}_j$ as calibration parameters. We have bM free calibration parameters, if we consider b maturities.

Beyond GPL: The GPCL model calibration

Notice that many $\widetilde{\Lambda}_j(t)$ will be equal to zero for all maturities, meaning that we can ignore their corresponding counting process $Z_j^2(t)$. One can think of deleting all the modes with jump sizes having zero intensity and keep only the nonzero intensity ones. Call $\alpha_1 < \alpha_2 < \ldots < \alpha_n$ the jump sizes with nonzero intensity. Then one renumbers progressively the intensities according to the nonzero increasing α: Z_j^2 becomes the jump of a cluster of size α_j.

The calibration procedure for GPCL is implemented using the α_j in the same way as for the GPL model. As concerns the GPCL intensities, in the tables we display $\binom{M}{\alpha_j}\widetilde{\Lambda}_j$, i.e. we multiply a cluster cumulated intensity for a given cluster size for the number of clusters with that size at time 0.

Beyond GPL: The GPCL model calibration

We also calibrate the GPL model, for comparison. To avoid confusion we denote the GPL cumulated intensities for the α_j mode by Λ_j^0, which reads, using the link with repeated defaults, as $\Lambda_j^0 = \binom{M}{\alpha_j}\widetilde{\Lambda}_j$. Given the arbitrary a-posteriori capping procedure in GPL, these $\widetilde{\Lambda}_j$ are not to be interpreted as cluster parameters, the only actual parameters being the Λ_j^0 directly, and they are to be interpreted as merely describing the pool counting process dynamic features.

Beyond GPL: The GPCL model calibration

More in detail, the optimal values for the amplitudes α_j in GPCL are selected, by adding non-zero amplitudes one by one, as follows, where typically $M = 125$:

1. set $\alpha_1 = 1$ and calibrate $\widetilde{\Lambda}_1$;
2. add the amplitude α_2 and find its best integer value by calibrating the cumulated intensities $\widetilde{\Lambda}_1$ and $\widetilde{\Lambda}_2$, starting from the previous value for $\widetilde{\Lambda}_1$ as a guess, for each value of α_2 in the range $[1, 125]$,
3. repeat the previous step for α_i with $i = 3$ and so on, by calibrating the cumulated intensities $\widetilde{\Lambda}_1, \dots, \widetilde{\Lambda}_i$, starting from the previously found $\widetilde{\Lambda}_1, \dots, \widetilde{\Lambda}_{i-1}$ as initial guess, until the calibration error is under a pre-fixed threshold or until the intensity $\widetilde{\Lambda}_i$ can be considered negligible.

Beyond GPL: The GPCL model calibration

The objective function f to be minimized in the calibration is the squared sum of the errors shown by the model to recover the tranche and index market quotes weighted by market bid-ask spreads:

$$f(\alpha, \widetilde{\Lambda}) = \sum_i \epsilon_i^2, \quad \epsilon_i = \frac{x_i(\alpha, \widetilde{\Lambda}) - x_i^{\text{Mid}}}{x_i^{\text{Bid}} - x_i^{\text{Ask}}} \tag{59}$$

where the x_i, with i running over the market quote set, are the index values R_0 for DJi-TRAXX index quotes, and either the index periodic premiums $R_0^{A,B}$ or the upfront premium rates $U^{A,B}$ for the DJi-TRAXX tranche quotes, see the appendix for more details.

Beyond GPL: The GPCL model calibration

The calibration data set is the DJi-TRAXX main series on the run on October, 2 2006. We have shown the table of the related quotes earlier, in the GPL calibration examples.

We calibrate three methods against such data set and we compare the results. They are listed in the tables.

1. The implied expected tranched loss method (hereafter IETL) described earlier .
2. The GPL model described earlier, i.e. $C_t = \min(Z_t, M)$ (referred to before as strategy 0). Such model, due to the capping feature, is not compatible with any of the previously described single-name dynamics avoiding repeated defaults.
3. The GPCL model described above (strategy 2), which represents an articulated solution to the repeated defaults problem. We implement the simplified version with cluster intensity $\widetilde{\lambda}_s$ depending only on cluster size $|s|$.

Beyond GPL: The GPCL model calibration

First, we check that there are no arbitrage opportunities on October, 2 2006, by calibrating the IETL method.

		Tranches					
Models	**Maturities**	0-3	3-6	6-9	9-12	12-22	22-100
ITL	3y	18.6%	0.2%	0.1%	0.0%	0.0%	0.0%
	5y	44.6%	4.2%	1.2%	0.6%	0.2%	0.1%
	7y	71.0%	14.5%	4.3%	2.1%	0.7%	0.2%
	10y	91.6%	49.2%	14.1%	6.4%	2.2%	0.4%
GPL	3y	18.6%	0.2%	0.1%	0.1%	0.0%	0.0%
	5y	44.5%	4.2%	1.2%	0.6%	0.2%	0.1%
	7y	70.8%	14.6%	4.3%	2.1%	0.7%	0.2%
	10y	91.2%	47.2%	14.6%	6.4%	2.2%	0.4%
GPCL	3y	18.7%	0.2%	0.1%	0.0%	0.0%	0.0%
	5y	44.7%	4.2%	1.2%	0.6%	0.2%	0.1%
	7y	70.9%	14.6%	4.3%	2.1%	0.7%	0.2%
	10y	91.2%	47.5%	14.5%	6.4%	2.2%	0.4%

Table 31: Implied expected tranched loss for the ITL, GPL and GPCL models. Results refer to DJi-TRAXX market.

Beyond GPL: The GPCL model calibration

Then we calibrate the GPL and GPCL models, and we obtain the calibration parameters presented in Table 32, while the expected tranched losses implied by these two models are included in Table 31. We point out that this is a joint calibration across tranche seniority and maturity, since we are calibrating all and every tranche and index quote with a single model specification. When looking at the outputs of the calibrated models on the different maturities, we see that both our models perform very well on maturities of 3 years, 5 years and 7 years, for which the calibration error is within the bid-ask spread. The 10 year maturity quotes are more difficult to recover, but both models are close to the market values, as we see from the left panel of Table 34. Notice, however, that the GPCL model has a lower calibration error ($10\% - 20\%$ better).

Beyond GPL: The GPCL model calibration

α_j	$\Lambda_j^0(T)$			
	3y	5y	7y	10y
1	0.778	1.318	3.320	4.261
3	0.128	0.536	0.581	1.566
15	0.000	0.004	0.024	0.024
19	0.000	0.007	0.011	0.028
32	0.000	0.000	0.000	0.007
79	0.000	0.000	0.003	0.003
120	0.000	0.002	0.003	0.008

α_j	$\binom{M}{\alpha_j}\widetilde{\Lambda}_j(T)$			
	3y	5y	7y	10y
1	0.882	1.234	3.223	3.661
3	0.128	0.615	0.682	1.963
15	0.001	0.002	0.023	0.023
19	0.000	0.009	0.016	0.043
57	0.000	0.000	0.002	0.007
80	0.000	0.000	0.000	0.010
125	0.001	0.005	0.042	0.042

Table 32: DJi-TRAXX pool. Left side: cumulated intensities, integrated up to tranche maturities, of the basic GPL model. Each row j corresponds to a different Poisson component with jump amplitude α_j. Right side: cumulated cluster intensities, integrated up to tranche maturities, and multiplied by the number of clusters of the same size at time 0. Each row j corresponds to a different cluster size α_j. The amplitudes/cluster-sizes not listed have an intensity below 10^{-7}. The recovery rate is 40%. All calibration errors within one bid-ask

Beyond GPL: The GPCL model calibration

We also apply the ITL, GPL and GPCL methods to the CDX index and tranches (see Table 28 for market quotes), following the same procedure used for the DJi-TRAXX above. We find better results, that are summarized in Table 33 and in the right panel of Table 34.

Beyond GPL: The GPCL model calibration

$\boldsymbol{\alpha_j}$	$\boldsymbol{\Lambda_j^0(T)}$			
	3y	5y	7y	10y
1	1.132	3.043	4.247	7.166
2	0.189	0.189	0.812	1.625
6	0.011	0.091	0.091	0.091
18	0.000	0.006	0.028	0.028
23	0.000	0.004	0.005	0.032
32	0.000	0.000	0.000	0.009
124	0.000	0.003	0.005	0.010

$\boldsymbol{\alpha_j}$	$\binom{M}{\alpha_j}\widetilde{\boldsymbol{\Lambda}}_{\boldsymbol{j}}(\boldsymbol{T})$			
	3y	5y	7y	10y
1	0.063	0.552	3.100	6.661
2	0.804	1.531	1.531	2.076
3	0.020	0.195	0.195	0.195
17	0.000	0.010	0.037	0.087
32	0.000	0.003	0.009	0.032
110	0.000	0.000	0.000	0.010
125	0.000	0.011	0.054	0.054

Table 33: CDX pool. Left side: cumulated intensities, integrated up to tranche maturities, of the basic GPL model. Each row j corresponds to a different Poisson component with jump amplitude α_j. Right side: cumulated cluster intensities, integrated up to tranche maturities, and multiplied by the number of clusters of the same size at time 0. Each row j corresponds to a different cluster size α_j. The amplitudes/cluster-sizes not listed have an intensity below 10^{-7}. The recovery rate is 40%.

Beyond GPL: The GPCL model calibration

	Att-Det	DJi-TRAXX 10y	
		GPL	GPCL
Index		0.00	0.00
Tranche	0-3	0.76	0.62
	3-6	-2.35	-1.93
	6-9	1.21	1.04
	9-12	-0.40	-0.36
	12-22	0.02	0.02
	22-100	0.00	0.00

	Att-Det	CDX 10y	
		GPL	GPCL
Index		0.00	-0.06
Tranche	0-3	1.43	1.60
	3-7	-0.45	-0.22
	7-10	0.22	0.25
	10-15	-0.08	-0.12
	15-30	0.01	0.07

Table 34: Calibration errors calculated with the GPL and GPCL models with respect to the bid-ask spread (i.e. ϵ_i in (59)) for tranches quoted by the market for the ten year maturity (see Tables 26 and 28). The left panel refers to DJi-TRAXX market quotes, while the right panel refers to CDX market quotes. Calibration errors for the other maturities are within the bid-ask spread and therefore they are not reported. The recovery rate is 40% .

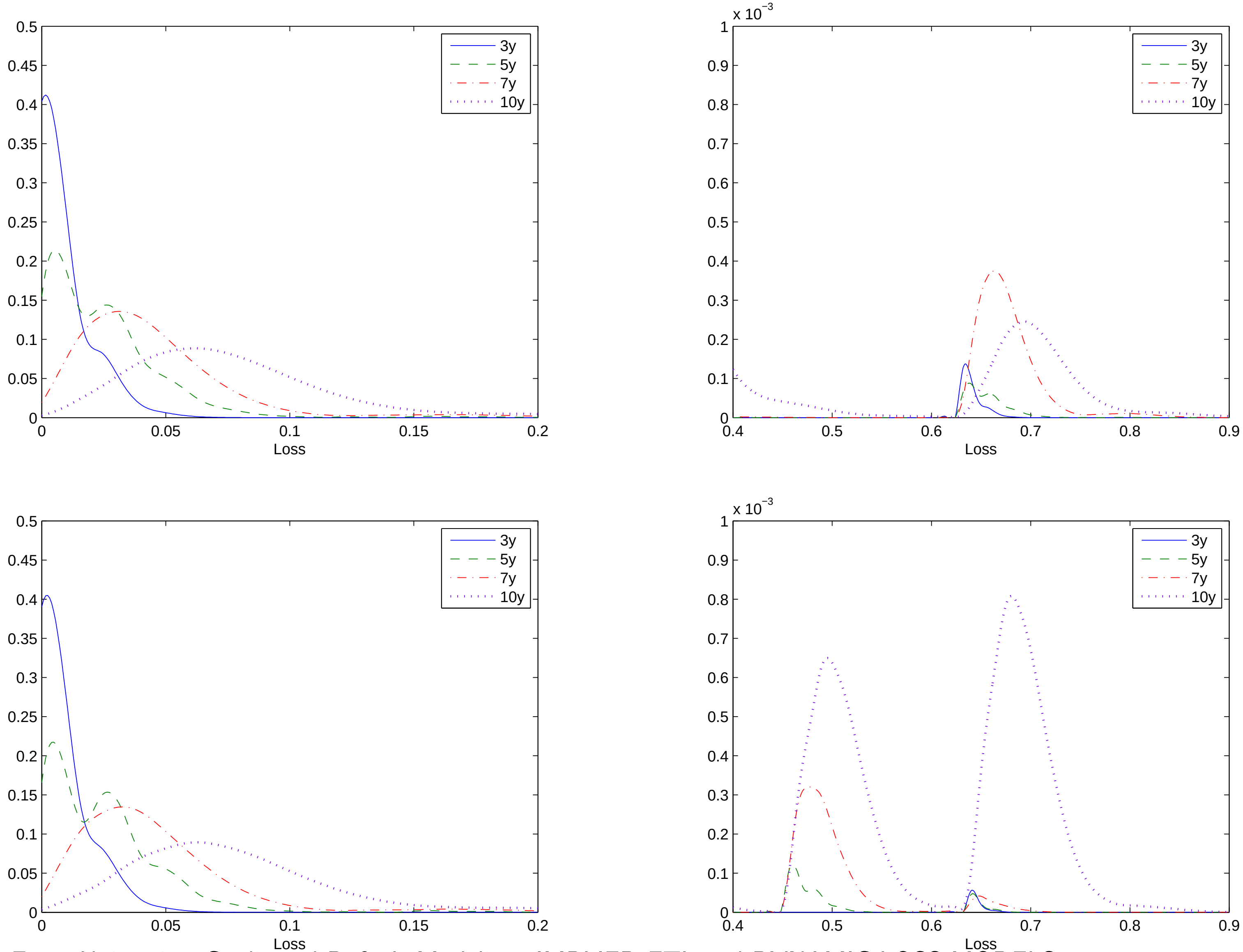


Figure 35:

Beyond GPL: The GPCL model calibration

Loss distribution evolution of the GPL model (upper panel) and of the GPCL model (lower panel) at all the quoted maturities up to ten years, drawn as a continuous line.

The probability distributions implied by the two dynamical models are similar at gross-grain view, as one can see in Figure 35, but they differ if we observe the fine structure. Indeed, the tails of the two distributions show different bumps. The GPCL model shows a more complex pattern, and, as one can see from Table 32, its highest mode is the maximum portfolio loss, while the GPL model has a less clear tail configuration.

Model Extensions: Spread dynamics

The valuation of credit index forward contracts or options maturing at time $T = T_a$ requires the calculation of the index spread at those future times, which in turn depends on the default intensity evolution.

Consider deterministic interest rates for an index whose default leg protects against losses in the index pool up to time T_b and where the spread premium payments occur at times $T_1, T_2, \ldots, T_b$.

We have the spread expression at T_a as

Model Extensions: Spread dynamics

$$R_T = \frac{\int_T^{T_b} D(T,t)\,\mathbb{E}_T[\,h_{\mathsf{Loss}}(t)\,]\;dt}{\sum_{i=1}^{b}\;\delta_i D(T,T_i)\left(1-\bar{C}_T-\int_T^{T_i}\mathbb{E}_T[\,h_{\bar{C}}(t)\,]\,dt\right)1_{\{T_i>T\}}}$$

The GPCL model presented in the previous sections has single-name and default counting intensities given by

$$h_{N_k^2}(t) = \sum_{s\ni\,k} J_s(t^-)\widetilde{\lambda}_s(t), \quad h_{Z_j^2}(t) = \sum_{|s|=j} J_s(t^-)\widetilde{\lambda}_s(t) \tag{60}$$

These intensities depend on which names have already defaulted. The dynamics of S_t can be enriched by more sophisticated modelling of $h_{\mathsf{Loss}}(t)$ and $h_{\bar{C}}(t)$, by explicitly adding stochasticity to the Poisson intensities $\widetilde{\lambda}_j(t)$, e.g. resorting to the Gamma, scenario or CIR extensions, similarly to what has been suggested for the GPL model.

Recovery dynamics

Recall the notion of recovery at default Rec_t:

$$d\text{Loss}_t = (1 - \text{Rec}_t)d\bar{C}_t \quad \text{(or, more precisely} \quad \text{Loss}_t = \int_0^t (1 - \text{Rec}_u)d\bar{C}_u). \quad (61)$$

Now we specify more about this notion. In general, for ease of computation, we assume Rec_t to be a $\mathcal{G}_t$-adapted and left-continuous (and hence predictable) process taking values in the interval $[0, 1]$. On predictability of the recovery process see also Bielecki and Rutkowski (2001).

Here $\mathcal{G}_t$ denotes the filtration consisting of default-free market information and of the default-count monitoring up to time t. This implies in particular, that the loss Loss_t is $\mathcal{G}_t$-adapted too, as is reasonable.

The no-arbitrage condition $d\text{Loss}_t \leq \ d\bar{C}_t$ is met if Rec_t takes values in $[0, 1]$. Equation (61) leaves us with the freedom of defining only two processes among Loss_t, $\bar{C}_t$ and Rec_t. The more natural approach would be modeling explicitly $(\bar{C}_t, \text{Rec}_t)$, obtaining Loss_t, or modeling explicitly $(\text{Loss}_t, \text{Rec}_t)$, obtaining $\bar{C}_t$, all of them adapted.

Recovery dynamics

However, if we choose to model both Loss_t and $\bar{C}_t$ as $\mathcal{G}_t$-adapted processes and to infer Rec_t, we have to ensure that the resulting process Rec_t implicit in $d\text{Loss}_t = (1 - \text{Rec}_t) d\bar{C}_t$ is indeed left-continuous (and hence $\mathcal{G}_t$-predictable).

Indeed, in some formulations the predictability of the recovery is not possible. It is also a notion not always realistic: whether one or 125 names default in instant $(t - dt, t]$ (i.e. $dC_t = 1$ or $dC_t = 125$, respectively), we would be imposing the recovery R_t to be the same in both cases and, in particular, to depend only on the information up to t^-.

What is interesting in spite of this unrealistic feature is that, under adapted-ness and left-continuity, the recovery rate can be expressed also in terms of the intensities of the loss and default rate processes. From equation $d\text{Loss}_t = (1 - \text{Rec}_t) d\bar{C}_t$, by definition of compensator, we obtain

$$\text{Rec}_t = 1 - \frac{h_{\text{Loss}}(t)}{h_{\bar{C}}(t)}. \tag{62}$$

Recovery dynamics

$$\mathsf{Rec}_t = 1 - \frac{h_{\mathsf{Loss}}(t)}{h_{\bar{C}}(t)}. \tag{63}$$

Equation (63) shows that the recovery rate at default is directly related to the intensities of both the loss and the default rate processes. Thus, the choice for the intensity dynamics does induce a dynamics for the recovery rate.

We now examine possible ways to model the loss more realistically, starting from a GPL or GPCL model formulated in terms of default counting process. This amounts to implicitly model the recovery rate, since the number of defaults and the loss are linked by the recovery at default.

Recovery dynamics through Deterministic mapping

A first approach to implicitly model recovery rates consists in defining the cumulated portfolio loss $\textsc{Loss}_t$ process as a deterministic function of the pool counting process $\bar{C}_t$ via a deterministic map. Set

$$\textsc{Loss}_t := \psi(\bar{C}_t),$$

where ψ is a non-decreasing deterministic function with $\psi(0) = 0$ and $\psi(1) \leq 1$. What does this imply in terms of recovery dynamics? We can easily write

$$d\textsc{Loss}_t = \sum_{k=1}^{M} \left[\frac{\psi(\bar{C}_{t-} + k/M) - \psi(\bar{C}_{t-})}{k/M} \right] 1_{\{d\bar{C}_t = k/M\}} d\bar{C}_t$$

which shows that the recovery at default in this case would not be predictable, depending explicitly from dC_t, except for very special ψ's.

Recovery dynamics through Deterministic mapping

A generalization based on a random process transformation (rather than a deterministic function) of the counting process leads to a more sophisticated implicit dynamics of the recovery process.

Consider a stochastic process $u \mapsto \Psi_u$ in time u, $\mathcal{G}_u$-adapted and taking values in $[0,1]$, right-continuous with left limit, and independent of the default counting process $\bar{C}_t$, and use it to map the positive non-decreasing pool counting process $\bar{C}_t$ taking values in $[0,1]$ into the portfolio cumulated loss Loss_t, sharing the same characteristics, i.e. define

$$\text{Loss}_t := \Psi_{\bar{C}_t}.$$

Further, assume (no-arbitrage conditions):

$$\Psi_0 = 0, \quad \Psi_1 \leq 1, \quad \text{and} \quad d\Psi_t \geq 0$$

This way the cumulated portfolio loss can be viewed as a stochastic time change of the process Ψ. Further, in order to allow for portfolio total loss, we enforce the stronger condition $\Psi_1 = 1$.

Recovery dynamics through Deterministic mapping

The time change does not spoil the analytical tractability of the model. If we know the probability distribution function of the pool counting process and of Ψ, we can simply derive the probability distribution function of the portfolio loss through an iterated expectation, thanks to independence:

$$\mathbb{Q}\{\mathsf{Loss}_t \leq x\} = \mathbb{E}\left[\, \mathbb{Q}\{\mathsf{Loss}_t \leq x | \bar{C}_t\} \,\right] = \int \mathbb{Q}\{\Psi_y \leq x\} p_{\bar{C}_t}(y) dy$$

Recovery dynamics through Deterministic mapping

As a relevant example, assume the process $u \mapsto \Psi_u$ is a Gamma process with shape parameter $\mu(u)$ and scale parameter ν. The monotonicity of the resulting loss process can be easily checked, while the probability distribution of the process can be calculated explicitly. Indeed, as a direct calculation can show, for any times $s < t < T$, the conditional distribution of Ψ_t, given Ψ_s and Ψ_T is known in terms of the Beta distribution.

The calculation of the unconditional distribution of the cumulated portfolio loss follows directly.

Exactly as for the previous case based on the deterministic transform ψ, here the implicit recovery at default turns out to be not predictable in general.

GPL and GPCL loss models: conclusions

We have extend the common Poisson shock (CPS) framework in two possible ways that avoid repeated defaults. The second way, more consistent with the original spirit of the CPS framework, leads to the Generalized-Poisson adjusted-Cluster-dynamics Loss model (GPCL).

We have illustrated the relationship of the GPCL with our earlier Generalized Poisson Loss (GPL) model, pointing out that while the GPCL model shares the good calibration power of the GPL model, it further allows for consistency with single names, thus constituing one of the few explict examples of top down approaches to loss modeling with real consistency for single names, or of bottom up approaches with real dynamical features.

GPL and GPCL loss models: conclusions

Further research concerns recovery dynamics, calibration and analysis of forward start tranches and tranche options, when liquid quotes will be available, and analysis of calibration stability through history.

A preliminary analysis of stability with the GPL model is however presented in Brigo, Pallavicini and Torresetti (2006b), showing good results. This is encouraging and leads to assuming the GPCL stability as well, although a rigorous check is in order in further work.