

# All your hedges in one basket

Leif Andersen, Jakob Sidenius and Susanta Basu present new techniques for single-tranche CDO sensitivity and hedge ratio calculations. Using factorisation of the copula correlation matrix, discretisation of the conditional loss distribution followed by a recursion-based probability calculation, and derivation of analytical formulas for deltas, they demonstrate a significant improvement in computational speeds

In a traditional synthetic collateralised debt obligation (CDO), the arranger tranches out credit losses on a pool of credit default swaps (CDSs) and passes them through to different investors. Assuming that investors for all tranches can be identified, the arranger is typically left with fairly moderate market exposure. For various reasons, placing the entire pool capital structure with investors has become increasingly difficult, and many recent credit basket derivatives expose the dealer to significant market risk. For instance, the recent 'single-tranche' CDO (STCDO) product involves the sale of a single CDO tranche to a single customer, leaving it to the arranger to manage the risk of the remaining capital structure. As STCDOs and similar 'custom' products offer significant customer benefits and are much less difficult to originate than traditional CDOs, such products are likely to increase in importance. This is especially true for managed trades where the customer has certain rights to alter the composition of the reference portfolio over time.

A basic prerequisite for active management of the risk of a credit basket derivative is the ability to accurately calculate the sensitivity of the security with respect to market and model parameters, most prominently the par CDS spreads of the underlying reference pool. The numbers of such sensitivities can be very large – many thousands – and can put considerable strain on computing resources. Moreover, the calculation of each of these derivatives can be significantly more challenging than calculating the price. For instance, in the popular Monte Carlo method, one often finds that the relative accuracy of CDS spread sensitivities can be hundreds of times lower than the relative price precision.

In this article, we discuss a number of practical techniques to improve and speed up sensitivity computations of CDOs, STCDOs and other credit basket securities. The discussion is broken into two parts, depending on the complexity of the security in question. In the main line of discussion, we tackle the class of relatively simple derivatives that allow for fast quasi-analytical approximations based on factor-reduction and discretisation of the portfolio loss distribution. As we shall show, CDOs and STCDOs belong to this class and allow for the application of extremely efficient recursions for the calculation of both prices and hedge parameters. Despite having received little attention in the literature so far, recursive algorithms for credit baskets appear to be increasingly popular with practitioners. We expose the method in detail here and back up its practical usage with efficient algorithms for loss unit discretisation and factor reduction of arbitrary correlation matrices.

In another line of discussion, we investigate whether techniques used for the quasi-analytical approximations are helpful in a Monte Carlo setting. Monte Carlo techniques are inherently slower than the quasi-analytical technique mentioned above, but can handle a larger number of exotic variations on the basic CDO/STCDO theme.<sup>1</sup>

## Notation and model

Consider a portfolio of credit default swaps on  $N$  different companies, each associated with a term curve of CDS par spreads and a recovery rate. For credit  $k$ , let  $I_k$  and  $R_k$  be the CDS notional and recovery rate per unit notional, respectively; all recovery rates are assumed constant for simplicity. Further, define  $p_k(T)$  as the risk-neutral default probability over  $[0, T]$  for credit  $k$ . With  $\tau_k$  denoting the random time of default for company  $k$ , we thus have:

$$p_k(T) = Q(\tau_k \leq T) \equiv 1 - e^{-\int_0^T \lambda_k(u) du} \quad (1)$$

where  $Q$  is the risk-neutral probability measure and  $\lambda_k$  is a (forward) default hazard rate function. The functions  $p_k(T)$ ,  $k = 1, \dots, N$  can be bootstrapped by standard means from the quoted CDS spreads and are assumed known for all  $T$ .

Equation (1) fully establishes the risk-neutral marginal distribution of each default time  $\tau_k$ . To construct the joint distribution of all default times, we here choose<sup>2</sup> to employ a Student- $t$  copula, which we quickly define for reference. Defining vectors  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)^T$  and  $\mathbf{T} = (T_1, \dots, T_N)^T$ , the joint default time distribution in the Student- $t$  copula, becomes:

$$Q(\boldsymbol{\tau} \leq \mathbf{T}) = t_{N,v} \left( t_{1,v}^{-1}(p_1(T_1)), \dots, t_{1,v}^{-1}(p_N(T_N)) \right) \quad (2)$$

where  $t_{1,v}$  and  $t_{N,v}$  are the one- and  $N$ -dimensional cumulative Student- $t$  distribution functions with  $v$  degrees of freedom, respectively. Recall that the density  $\eta_{N,v}$  of an  $N$ -dimensional Student- $t$  distribution with correlation matrix  $\Sigma$  is:

$$\eta_{N,v}(\mathbf{z}) = C_{N,v} \left( 1 + v^{-1} \mathbf{z}^T \Sigma^{-1} \mathbf{z} \right)^{-\frac{v+N}{2}}, \quad C_{N,v} \equiv \frac{\Gamma\left(\frac{v+N}{2}\right)}{\Gamma\left(\frac{v}{2}\right) \sqrt{|\Sigma|} (v\pi)^N} \quad (3)$$

where  $\Gamma$  is the gamma function. For high degrees of freedom, (3) approaches a Gaussian distribution and the copula (2) becomes the popular Gaussian copula, the *de facto* market standard model (see, for instance, Schmidt & Ward, 2002). For finite degrees of freedom, the  $t$ -distribution has fatter tails than the Gaussian distribution and is known to generate tail dependence in the joint distribution. Econometric evidence seems to support such features (see Mashal & Naldi, 2002). For later use, recall that if  $\mathbf{Y}$  is an  $N$ -dimensional standard Gaussian variable with correlation matrix  $\Sigma$  and  $g$  is a scalar chi-square distributed random variable with  $v$  degrees of freedom, then:

$$\mathbf{Z} = \mathbf{Y} \sqrt{v/g} \quad (4)$$

follows a  $N$ -dimensional Student- $t$  distribution with correlation matrix  $\Sigma$  and density (3).

According to (4), a Student- $t$  variable can be interpreted as a Gaussian variable when conditioned on the state of the chi-square scalar variable  $g$ . While the Student- $t$  copula will be sufficient for the purposes of this article, we notice that all the techniques discussed would hold unchanged if the distribution of  $g$  is altered to something other than chi-square. In fact, our central recursion algorithm (see below) can be used to calculate the loss distribution for any copula with conditionally independent default times. Frey & McNeil (2001) contains details about the Student- $t$  and many other copulas, including general 'mixed Gaussian' copulas.

At its most abstract level, a credit basket security is simply a series of payouts given by a functional dependence on the realised value of  $\boldsymbol{\tau}$ . Letting this function be denoted  $f$ , the time 0 value of the security can, by standard arbitrage theory, be written as:

$$V(0) = E(f(\boldsymbol{\tau})) \quad (5)$$

<sup>1</sup> One interesting such variation is the construction of single-tranche baskets of STCDOs. This 'baskets-in-a-basket' variation is difficult (although not impossible) to handle in a portfolio loss setting but easily priced by Monte Carlo

<sup>2</sup> This choice is for illustrative purposes and reflects the widespread usage of the Student- $t$  copula, rather than any particular technical or empirical merit of this copula. Extensions to the more stationary Archimedean copulas are straightforward

where  $E(\cdot)$  is the risk-neutral expectation operator. Notice that we implicitly assume that  $f$  contains the necessary discounting of future cashflows.

With our assumption (2) about the distribution of  $\boldsymbol{\tau}$ , the expectation in (5) can be written explicitly. Specifically, differentiation of (2) produces the density  $\boldsymbol{\varphi}$  of the default time vector  $\boldsymbol{\tau}$ :

$$\boldsymbol{\varphi}(\mathbf{T}) = C_{N,v} \left( \mathbf{1} + \mathbf{v}^{-1} \mathbf{z}^T \boldsymbol{\Sigma}^{-1} \mathbf{z} \right)^{-\frac{v+N}{2}} \prod_{k=1}^N \lambda_k(T_k) (1 - p_k(T_k)) \eta_{1,v}(z_k)^{-1} \quad (6)$$

where  $z_k \equiv t_{1,v}^{-1}(p_k(T_k))$ . From (5), then:

$$V(0) = \int_{\mathbb{R}_+^N} f(\mathbf{T}) \boldsymbol{\varphi}(\mathbf{T}) d\mathbf{T} \quad (7)$$

As it stands, (7) is an  $N$ -dimensional integral over all default times and, for a general payout, will have to be evaluated by Monte Carlo methods. However, in a number of cases of practical importance, the complexity of the problem can be reduced significantly by recasting the payout in terms of cumulative portfolio losses. To provide some notation for this idea (for which we shall provide details later), define the aggregate portfolio loss  $\Lambda(t)$  as the sum of all credit losses on horizon  $[0, t]$ ,  $\Lambda(t) = \sum_{\tau_k \leq t} I_k(1 - R_k)$ . The distribution of  $\Lambda(t)$  is obviously discrete, with a finite number of possible outcomes between zero and  $\Lambda_{\max} = \sum_{k=1}^N I_k(1 - R_k)$ .

As it turns out, the price of many credit basket derivatives can be related directly to expectations of the form:

$$E(A(\Lambda(t))) \quad (8)$$

where  $A$  is some known function. Examples of this will be given shortly. Calculation of (8) involves construction of the discrete distribution of  $\Lambda(t)$ . For tractability, it is often useful to coarsen the loss distribution through a loss unit,  $u$ . Then, the  $k$ th company in the reference pool is assigned a (non-negative integer) loss weight,  $w_k$ , calculated by rounding  $(1 - R_k)I_k/u$  to the nearest integer. The maximal loss  $\Lambda_{\max}$  is  $u \times l_{\max}$ , where  $l_{\max}$  is the sum of all the loss weights. This discretisation will in general lead to rounding errors (except in simple cases such as, for example, when  $I_k(1 - R_k)$  is independent of  $k$ ) and the loss unit will have to be chosen small enough to keep these tolerable. On the other hand, the total computational effort is inversely proportional to the loss unit so a compromise must be found. See the Appendix for one approach to this.

With the loss unit procedure, we have discrete loss probabilities  $P(l; t)$ ,  $l = 0, 1, \dots, l_{\max}$  and (8) is approximated by:

$$E(A(\Lambda(t))) = \sum_{l=0}^{l_{\max}} A(ul) P(l; t) \quad (9)$$

If defaults of the individual companies in the reference basket are independent, the construction of the loss distribution is most efficiently accomplished by the following recursive argument (see Pykhtin & Dev, 2003, for a special case of the recursion). Suppose we know the loss distribution  $P^K(l; t)$ ,  $l = 0, \dots, l_{\max, K}$  for a reference pool of some size  $K \geq 0$ . Now suppose we add another company to the pool with loss weight  $w_{K+1}$  and known default probability  $p_{K+1}(t)$ . Then using the independence of defaults we find for the loss distribution of the larger basket:

$$P^{K+1}(l; t) = P^K(l; t)(1 - p_{K+1}(t)) + P^K(l - w_{K+1}) p_{K+1}(t), l = 0, \dots, l_{\max, K} + w_{K+1} \quad (10)$$

This recursive relation can be used to build the loss distribution  $P(l; t) = P^N(l; t)$  from the boundary case of the empty basket  $P^0(l; t) = \delta_{0,l}$ , where Kronecker's delta  $\delta_{i,j}$  is one for  $i = j$  and zero otherwise.

Note that the asymptotic complexity order of the recursion is  $O(N \times l_{\max})$ . Since an increase in basket size will usually lead to about the same relative increase in the maximal loss, the cost of building the conditional loss distribution grows as roughly the square of the basket size. One might think that this could be improved by using Fourier transform methods (see, for example, Gregory & Laurent, 2003, and Merino & Nyfeler, 2002). However, due to the computational burden associated with evaluating the char-

acteristic function, this turns out to often be significantly slower than the recursion. For the examples in this article, we found that the Fourier approach (with code taken from the IMSL C standard library) involved roughly five to 25 times more computing time than the approach shown here.

In the Student- $t$  copula framework, we obviously do not in general have the independence of defaults required for (10). As we shall discuss below, however, often we can approximate the co-dependence structure by one of *conditional* independence.

### Risk management of credit baskets

As is the case for all derivatives, risk management of credit basket derivatives is a matter of combining the basket portfolio with a dynamic portfolio of plain vanilla securities (CDSs, coupon bonds, asset swaps, etc) such that overall sensitivities to key exposure variables are within acceptable bounds. For a credit basket, the primary exposures are obviously to default-related variables such as the CDS credit spreads and recovery rates of the individual companies in the CDS pool. Of these, the former is typically the most important for investment-grade companies in the pool, with the latter gaining in importance as credit spreads increase, and the trader needs to deal with the likelihood of having to physically settle the CDS.

Of the key hedge parameters, the credit spread sensitivities are generally the most challenging to calculate numerically and shall consequently receive most of our attention. We note that credit spreads enter the valuation formula (7) through the dependence of (6) on the hazard rate curves  $\lambda_k(\cdot)$ , which again are bootstrapped out of the CDS credit spread quotes observed in the market. By applications of the chain-rule, we can always convert hazard rate sensitivities into credit spread sensitivities; we shall focus solely on the former here. For flexibility, we here define sensitivities to hazard rates as Gateaux (functional) derivatives with respect to arbitrary perturbation functions. For instance, consider a hazard rate curve  $k$  and define a deterministic perturbation function  $\xi(t)$ . Then we are interested in calculating the derivative:

$$\partial_{k,\xi} V \equiv \frac{d}{d\boldsymbol{\mu}} V(0; \lambda_1, \dots, \lambda_k + \boldsymbol{\mu} \boldsymbol{\xi}_k, \dots, \lambda_N) \Big|_{\boldsymbol{\mu}=0} = \frac{d}{d\boldsymbol{\mu}} V(0; \boldsymbol{\lambda} + \boldsymbol{\mu} \boldsymbol{\xi}_k) \Big|_{\boldsymbol{\mu}=0} \quad (11)$$

where  $\boldsymbol{\mu}$  is a scalar and  $\mathbf{e}_k$  is the  $k$ th natural basis vector in  $\mathbb{R}^N$ . To convert such quantities to a finite number  $n$  of derivatives with respect to credit spreads at various maturities, we would need to calculate the Gateaux derivatives for exactly  $n$  different spanning perturbation functions  $\xi_j(t)$ ,  $j = 1, \dots, n$ . These could, for instance, be chosen to be piecewise flat:  $\xi_j(t) = 1_{t < T_j}$ .

In practice, due to jumps/gaps in credit spreads and to high market friction, which makes frequent rehedging impractical, a successful hedging strategy may take into account the sensitivities to finite-size changes in spreads. Such sensitivities can be calculated efficiently with the techniques presented in this article, although for brevity we have chosen to focus on the calculation of spread deltas. Another important risk factor is the basket correlation matrix. This should ideally be implied from the market, but this is not (yet) practical, and it is therefore important to be able to calculate the impact of a possible misspecification. Our techniques rely only on generic properties of basket correlation matrices and so are ideally suited – in conjunction with the speed of valuation – to allow calculation of implied correlation as well as the impact of correlation changes.

### Factor reduction of correlation matrices

It has been noted in many recent articles (see, for example, Merino & Nyfeler, 2002, and Frey & McNeil, 2001) that specialising to correlation matrices with a factor structure can yield significant improvements in speed for many basket default securities. In this section, we shall briefly review the concept of factor structure, and then move on to the important practical question of how to calculate an optimal factor structure approximation to any correlation matrix.

Going back to equation (4), we recall that a central piece of our model is an  $N$ -dimensional standard Gaussian vector  $\mathbf{Y}$  with correlation matrix  $\boldsymbol{\Sigma}$ . In

practice,  $\Sigma$  is typically estimated by factor analysis or regression of equity returns. We say that  $\Sigma$  has an  $M$ -dimensional factor structure if we can write<sup>3</sup>:

$$\mathbf{Y} = \mathbf{c}\mathbf{X} + \mathbf{e} \quad (12)$$

where  $\mathbf{X}$  is an  $M$ -dimensional vector of independent standard Gaussian variables (the 'factors'),  $\mathbf{c}$  is an  $N \times M$  loading matrix and  $\mathbf{e}$  is an  $N$ -dimensional vector of independent zero-mean Gaussian variables with variances less than one. Our hope is that  $M \ll N$ , either exactly or to a good degree of accuracy. For the empirical correlation matrices one encounters in credit basket applications, we typically find that  $M = 1$  or  $2$  provides sufficient accuracy. In rare cases – particularly for small baskets – it might be necessary to use  $M = 3$  or higher.

Let us turn to the problem of calculating, for given  $M$ , a weighting matrix  $\mathbf{c}$  such that the correlation matrix generated by the right-hand side of (12) is as close as possible to a given  $\Sigma$ . Using a least-squares (Frobenius) norm on all non-diagonal elements, this problem reduces to solving the minimisation problem:

$$\min_{\mathbf{c}} \text{tr}(\Sigma - \mathbf{c}\mathbf{c}^T - \mathbf{F})(\Sigma - \mathbf{c}\mathbf{c}^T - \mathbf{F})^T \quad (13)$$

where  $\text{tr}$  is the usual matrix trace operator (sum of diagonal) and  $\mathbf{F}$  is a diagonal matrix ensuring that the diagonal of the factor-reduced correlation matrix is one throughout, that is:

$$\text{diag } \mathbf{F} = \mathbf{1} - \text{diag } \mathbf{c}\mathbf{c}^T \quad (14)$$

Note that (12) implies that all diagonal elements of  $\mathbf{c}\mathbf{c}^T$  are less than or equal to one and we must therefore solve (13) with this constraint.

Solution of (13) can, in principle, be obtained by brute force through the application of a multi-dimensional constrained optimisation algorithm. For large baskets, however, the dimension of the optimisation problem can exceed 1,000, making such methods impractical. Instead, we can use the fact that for a given  $\mathbf{F}$ , the solution to (13) can be found by principal components analysis (PCA). Specifically, let  $\mathbf{E}$  be the matrix of normalised (column) eigenvectors of  $\Sigma - \mathbf{F}$ ; then the solution to (13) is  $\mathbf{c} = \mathbf{E}\sqrt{\Lambda_M}$  where  $\Lambda_M$  is a diagonal matrix containing the  $M$  largest eigenvalues of  $\Sigma - \mathbf{F}$ . The solution found this way, however, does generally not satisfy (14), so we perform an iteration  $i = 0, 1, 2, \dots$  where in each step we:

- i) perform a PCA decomposition of  $\Sigma - \mathbf{F}^{(i)}$ , yielding  $\mathbf{c}^{(i)} = \mathbf{E}^{(i)}\sqrt{\Lambda_M^{(i)}}$ .
- ii) calculate  $\mathbf{F}^{(i+1)}$  from (14).
- iii) stop when  $\mathbf{F}^{(i+1)}$  is sufficiently close to  $\mathbf{F}^{(i)}$  (in a least-squares sense).

It is not difficult to show that, for an arbitrary starting guess of  $\mathbf{F}$ ,  $\text{tr}(\mathbf{F}^{(i+1)} - \mathbf{F}^{(i)})(\mathbf{F}^{(i+1)} - \mathbf{F}^{(i)})^T \rightarrow 0$  for large  $i$ , as desired. In practice, only a few iterations are necessary. We note that the algorithm above does not automatically guard against the constraint that all diagonal elements of  $\mathbf{c}\mathbf{c}^T$  are strictly less than one. While the algorithm above will rarely violate the constraint in practice, if necessary the constraint can be satisfied by suitably adjusting the number of factors.

### STCDO pricing from loss distribution

We now give an example of the portfolio loss approach discussed earlier by applying it to the pricing of STCDOs (and, by extension, CDOs). To first properly define the STCDO payout, we define a loss tranche of a pool of CDSs as an interval  $[L, H]$ . The floating (or 'asset' or 'option') leg of the STCDO pays the default losses in this interval, that is, it starts paying whenever the basket losses  $\Lambda$  exceed  $L$  and can pay at most  $H - L$  (when losses exceed  $H$ ). Losses are paid when they occur. The fixed (or 'coupon' or 'premium') leg pays a periodic coupon calculated as a fixed rate on the (stochastically decaying) tranche notional initially set to  $H - L$ .

Let us first consider the floating leg. Although, strictly speaking, its cashflows are timed by the stochastic default times, for simplicity we shall 'bucket' the defaults into time slots coinciding with the coupon periods for the fixed leg. We then calculate an effective cashflow by assuming that defaults always occur (and cashflows are paid) at the period mid-points. Let  $T_i$ ,  $i = 1, \dots, n$  be the  $i$ th strictly future coupon date and let  $P_i(x)$  be the basket loss distribution for horizon  $T_i$ . Further, let  $T_0 = 0$  and set  $P_0(x) = \delta(x)$ . Then

## A. CPU times for factor computation

|       | N = 25 | N = 50 | N = 100 | N = 200 |
|-------|--------|--------|---------|---------|
| M = 1 | 0.01   | 0.09   | 0.55    | 4.32    |
| M = 2 | 0.06   | 0.23   | 1.26    | 7.42    |
| M = 3 | 0.09   | 0.34   | 1.80    | 10.49   |

Notes: CPU times for factor computations on historically estimated correlation matrices. All times in seconds on a 1GHz Pentium IV processor

the expected loss to be absorbed by the tranche up to time  $T_i$  is:

$$e_i \equiv E\left(\left[\min(\Lambda(T_i), H) - L\right]^+\right) \quad (15)$$

which we recognise as being of the form (8). The expected loss over the  $i$ th coupon interval  $[T_{i-1}, T_i]$ , that is, the floating leg cashflow associated with the midpoint  $\bar{T}_i \equiv \frac{1}{2}(T_{i-1} + T_i)$ , is simply  $e_i - e_{i-1}$ .

For the fixed leg, let  $\theta_i$  and  $\alpha_i$  be the coupon rate and accrual factor, respectively, for the  $i$ th coupon and let  $I_{\text{tranche}}(T_i)$  be the tranche notional at time  $T_i$  (so that  $I_{\text{tranche}}(T_0) = H - L$ ). Then by assuming that losses are evenly distributed over the coupon period, the  $i$ th coupon cashflow paid at time  $T_i$  is:

$$\begin{aligned} & \alpha_i \theta_i E\left[\frac{1}{2}(I_{\text{tranche}}(T_{i-1}) + I_{\text{tranche}}(T_i))\right] = \\ & \alpha_i \theta_i \frac{1}{2}(I_{\text{tranche}}(T_0) - e_{i-1} + I_{\text{tranche}}(T_0) - e_i) = \\ & \alpha_i \theta_i (H - L - \frac{1}{2}(e_{i-1} + e_i)) \end{aligned}$$

Introducing a time zero zero-coupon curve  $D(\cdot)$ , we can now finally write the time zero value of the STCDO (as seen from the fixed leg receiver) as:

$$V(0) = \sum_{i=1}^n \alpha_i \theta_i D(T_i) (H - L - \frac{1}{2}(e_i + e_{i-1})) - \sum_{i=1}^n D(\bar{T}_i) (e_i - e_{i-1}) \quad (16)$$

To calculate (16), we now turn to the problem of estimating the quantities  $e_i$  defined in (15). From equation (9) we get:

$$\begin{aligned} e_i &= \sum_{l=l_i+1}^{l_{\max}} (\min\{lu, H\} - L) P(l; T_i) = \\ & \sum_{l=l_i+1}^{l_h} (lu - L) P(l; T_i) + (H - L) \sum_{l=l_h+1}^{l_{\max}} P(l; T_i) \end{aligned} \quad (17)$$

where<sup>4</sup>  $l_i := \min(l_{\max}, \lceil Lu \rceil)$  and  $l_h = \min(l_{\max}, \lceil Hu \rceil)$ . It now only remains to calculate the loss probabilities  $P(l; T_i)$ .

Assume now that we are working in a Student- $t$  copula where the correlation matrix has an  $M$ -dimensional factor structure of the form (12). Define  $\Omega = (\mathbf{X}, g)$  where  $g$  is the scalar random variable in (4). The probability of default of the  $k$ th company in the CDS pool conditioned on  $\Omega$  is given by:

$$p_k(t|\Omega) = \Phi\left(\frac{\sqrt{\frac{g}{v}} t_{1,v}^{-1}(p_k(t)) - \mathbf{c}_k \mathbf{X}}{\sqrt{1 - \mathbf{c}_k \mathbf{c}_k^T}}\right) \quad (18)$$

where  $\mathbf{c}_k$  are the relevant factor loadings (=  $k$ th row of the load matrix  $\mathbf{c}$  in (12)), and  $\Phi$  is the cumulative Gaussian distribution function.

Now let  $P(l; t|\Omega)$  be the portfolio loss distribution conditioned on  $\Omega$ . Given  $\Omega$ , all companies in the pool are easily seen from (4) and (12) to be independent, allowing us to use the recursion formula (10) for  $P(l; t|\Omega)$  once we substitute  $p_k(t|\Omega)$  for  $p_k(t)$ . The unconditional loss distribution can

<sup>3</sup> Note that  $M$  is unrelated to the rank of the correlation matrix and that approximating  $\Sigma$  by a factor structure (as we shall do below) cannot be accomplished simply by a truncated eigenexpansion (principal components analysis). The key here is the fact that equation (12) allows for arbitrary residuals, whereas a principal components analysis would set all residuals to zero

<sup>4</sup> In our notation,  $[x]$  denotes the integer part of the real number  $x$ . Note that  $[x] \leq x$

then be calculated by integrating out  $\Omega$ :

$$P(l;t) = \int_{\mathbb{R}^M \times \mathbb{R}_+} P(l;t|\Omega)q(\Omega)d\Omega$$

where the joint density of  $\Omega$ ,  $q(\Omega)$ , is simply the product of  $M$  standard Gaussian densities and the one-dimensional chi-square density with  $v$  degrees of freedom. This  $M + 1$ -dimensional integral can easily be done by Gaussian quadrature, after which we can complete, as desired, the calculation of equation (17). The use of Gaussian quadrature implies that the computational workload is multiplied for each additional correlation factor by the number of integration nodes for that factor. Fortunately, it turns out that the number of nodes required for good accuracy decreases with the number of factors.

Finally, notice that the popular Gaussian copula is nested inside Student- $t$  copula. To recover the Gaussian copula, in equation (18) replace  $t_{1,v}^{-1}$  with  $\Phi^{-1}$  and fix  $g = v$ . In the calculation of  $P(l)$  we can then omit the integration over  $g$ , making the Gaussian copula faster to evaluate than the Student- $t$  copula.

### Sensitivity calculations in the portfolio loss framework

At the heart of the technique used in the previous section is the evaluation of conditional expectations of the form:

$$E(A|\Omega) = \sum_{l=0}^{l_{\max}} A(l)P(l|\Omega) \quad (19)$$

where we have suppressed the time argument for simplicity. Calculating sensitivities of basket derivatives in the portfolio loss approach ultimately amounts to calculating sensitivities of the expectation above, a problem to which we now turn.

Specifically, consider the calculation of the Gateaux hazard rate sensitivities (defined earlier) of (19) for  $\lambda \mapsto \lambda + \mu \xi e_k$ , for a fixed value of  $k$ . We write:

$$\left. \frac{\partial E(A|\Omega)}{\partial \mu} \right|_{\mu=0} = \left. \frac{dp_k(\Omega)}{d\mu} \right|_{\mu=0} \frac{\partial E(A|\Omega)}{\partial p_k(\Omega)} = \left. \frac{dp_k(\Omega)}{d\mu} \right|_{\mu=0} \sum_{l=0}^{l_{\max}} A(l) \frac{\partial P(l|\Omega)}{\partial p_k(\Omega)} \quad (20)$$

where we have used the fact that  $\mu$  will not affect  $p_j(\Omega)$  for  $j \neq k$ . We calculate (20) in two steps corresponding to the two factors on the right-hand side. First note that since  $dp_k(\Omega)/dp_j = 0$ , for  $j \neq k$ , we have  $dp_k(\Omega)/d\mu = dp_k(\Omega)/dp_k \times dp_k/d\mu$ , where:

$$\left. \frac{dp_k}{d\mu} \right|_{\mu=0} = (1 - p_k) \int_0^T \xi(t) dt$$

This equation follows from the fact that a shock  $\mu \xi(t)$  to the  $k$ th hazard rate will multiply the survival probability  $q_k \equiv 1 - p_k$  by a factor  $\exp(-\mu \int_0^T \xi(t) dt)$ . From equation (18), we also get by a direct calculation:

$$\left. \frac{dp_k(\Omega)}{d\mu} \right|_{\mu=0} = \frac{g(1-p_k)B\left(\frac{1}{2}, \frac{v}{2}\right) \int_0^T \xi(t) dt}{\sqrt{2\pi} \sqrt{1 - \mathbf{c}_k \mathbf{c}_k^T}} \left( \frac{v + (t_{1,v}^{-1}(p_k))^2}{v} \right)^{\frac{v+1}{2}} \exp\left(-\frac{1}{2} \frac{\left(\sqrt{\frac{g}{v}} t_{1,v}^{-1}(p_k) - \mathbf{c}_k \mathbf{X}\right)^2}{1 - \mathbf{c}_k \mathbf{c}_k^T}\right) \quad (21)$$

where  $B(x, y)$ , is the classic beta function.

Turning to the calculation of  $\partial P(l|\Omega)/\partial p_k(\Omega)$  in (20), we now fix  $k$  and consider the reduced basket obtained by removing the  $k$ th name from the original basket. Letting  $P^{(k)}(l|\Omega)$  be the conditional loss distribution for the reduced basket, we have:

$$P(l|\Omega) = P^{(k)}(l|\Omega)(1 - p_k(\Omega)) + P^{(k)}(l - w_k|\Omega)p_k(\Omega) \quad (22)$$

From this equation it is trivial to directly determine the distribution  $P^{(k)}(l|\Omega)$  and, in turn, we can use this to calculate the partial derivative:

## B. CPU times for quasi-analytical STCDO pricing

|           |       | Value only |         |         | Value and sensitivities |         |         |
|-----------|-------|------------|---------|---------|-------------------------|---------|---------|
|           |       | N = 50     | N = 100 | N = 200 | N = 50                  | N = 100 | N = 200 |
| Gaussian  | M = 1 | 0.02       | 0.04    | 0.17    | 0.30                    | 0.64    | 1.63    |
|           | M = 2 | 0.10       | 0.25    | 0.92    | 0.42                    | 1.06    | 3.81    |
| Student-t | M = 1 | 0.08       | 0.24    | 0.94    | 0.42                    | 1.18    | 4.14    |
|           | M = 2 | 0.36       | 1.16    | 4.67    | 1.01                    | 3.74    | 16.12   |

Notes: CPU times for computations on actual STCDO deals. All times are in seconds on a 1GHz Pentium IV processor. No simplifying assumptions were made for credit spreads, recovery rates, correlations or deal structure. The underlying deal maturity is five years, on a quarterly schedule. Computed sensitivities in the last three columns are parallel shift hazard rate sensitivities, default positions and recovery rate sensitivities for all credits

$$\frac{\partial P(l|\Omega)}{\partial p_k(\Omega)} = P^{(k)}(l - w_k|\Omega) - P^{(k)}(l|\Omega) \quad (23)$$

With (23) and (21) we can calculate (20), after which the usual integration over the risk factors  $\Omega$  will yield the desired Gateaux derivative  $dE(A)/d\mu$ .

The techniques outlined above can easily be applied to gauge recovery sensitivities and the effects of sudden default of individual names (the 'default positions'). For instance, to calculate default positions, we simply remove company  $k$  from the loss pool as in (22), set  $p_k(t|\Omega) \equiv 1$ , and reconstitute the full loss distribution portfolio. This 'trick' of removing a single name (or several names) from the loss distribution is very useful and can be applied to more general scenario analysis. For instance, it is straightforward to find 'gamma' and 'cross-gamma' effects by calculating the effect of simultaneous finite-size shocks to hazard rates.

We should point out that the ease and speed with which various sensitivities can be calculated and derived in the recursion-based framework is a significant practical advantage of this method over competing algorithms. To illustrate the computational speed of our quasi-analytical approximation, table B shows CPU times for some computations on actual STCDO deals.

### Monte Carlo simulations

We now move on to more complicated basket derivatives where the portfolio loss framework cannot be used. This is the case, say, for the 'nested' CDOs mentioned in footnote 1 and for CDOs with collection accounts and other complications. For this, we wish to use Monte Carlo to evaluate the expectation (5) and, more importantly, the derivative (11). In this setting, we want to investigate whether two key techniques introduced in the previous section are useful: factor decomposition of the correlation matrix and forming the derivative of the loss density in the delta calculation.

Assuming that the correlation matrix of the Student- $t$  copula has an  $M$ -dimensional factor structure, a basic Monte Carlo algorithm takes the form:

- i) draw an  $M + N$ -dimensional sample  $\mathbf{U}$  of uncorrelated standard Gaussian numbers. Let the first  $M$  numbers represent  $\mathbf{X}$  in equation (12); let the remaining numbers be denoted  $\mathbf{E}$ .
- ii) generate the correlated sample  $\mathbf{Y}$  as  $\mathbf{Y} = \mathbf{A}\mathbf{U}$ ,  $\mathbf{U} = [\mathbf{X}^T \mathbf{E}^T]^T$ , where  $\mathbf{A} \equiv [\mathbf{c} \sqrt{\mathbf{F}}]$  is an  $N \times (N + M)$  matrix. (See (14) for the definition of the diagonal matrix  $\mathbf{F}$ .)
- iii) draw a single chi-square sample  $g$  with  $v$  degrees of freedom to generate a correlated Student- $t$  sample vector  $\mathbf{Z}$  from equation (4).
- iv) calculate the random default time vector  $\boldsymbol{\tau}$  by  $\tau_k = p_k^{-1}(t_{1,v}(Z_i))$ ,  $k = 1, \dots, N$ .
- v) evaluate the payout function  $f$ , including discounting.

Steps i-v would be repeated a total of  $m$  times and the average of  $f$  would be used as the price estimate. The convergence of the algorithm is  $O(m^{-1/2})$ .

<sup>5</sup> For efficiency reasons, the map between  $Z_i$  and  $\tau_i$  obviously should be pre-tabulated on discrete grids outside the Monte Carlo loop

As the matrix  $\mathbf{F}$  is diagonal, the workload of steps i–ii is  $O(NM)$ , down from the  $O(N^2)$  operations needed for general correlation matrices where step ii would involve a multiplication with a non-diagonal Cholesky matrix. For  $N$  of the magnitude of 100–200 or more, this reduction can obviously be significant, although the overall effort of the Monte Carlo algorithm is often dominated by the, typically complex, payout evaluation in step v.

Consider now calculating the Gateaux derivative (11). In a brute-force approach, this derivative is calculated by finite differencing:

$$\partial_{k,\xi} V \approx \frac{V(0; \boldsymbol{\lambda} + \Delta \xi \mathbf{e}_k) - V(0; \boldsymbol{\lambda})}{\Delta}$$

where  $\Delta$  is some finite scalar. In the Monte Carlo algorithm, we would need to insert the following two steps<sup>6</sup> for each index  $k$  for which we want to evaluate the derivative:

□ vi) perturb the  $k$ th default time  $\tau_k$  to  $\tilde{\tau}_k = \tilde{p}_k^{-1}(t_{1,v}(Z_k))$ , where  $\tilde{p}_k(t) \equiv 1 - \exp(-\int_0^t [\lambda_k(u) + \Delta \xi(u)] du)$ .

□ vii) compute  $\delta_{k,\Delta} = [f(\tau_1, \dots, \tau_k, \dots, \tau_N) - f(\tau_1, \dots, \tilde{\tau}_k, \dots, \tau_N)]/\Delta$ .

The average of  $\delta_{k,\Delta}$  over the  $m$  Monte Carlo paths would form the estimate of the Gateaux derivative.

The scheme above is simple, but sometimes problematic. First, the determination of  $\Delta$  can be tricky: if picked too low in absolute magnitude, statistical noise of the derivative estimator will overwhelm the result; if picked too high, convexity effects ('gamma') can make the derivative estimate biased and unreliable. Second, the fact that the algorithm requires a re-evaluation of the payout routine in step vii leads to a potentially enormous amount of payout function calls: to calculate  $n$  different Gateaux sensitivities for all  $N$  companies in the pool would involve  $O(N \times m \times n)$  payout evaluations. For complicated payout functions – which are the primary candidates for Monte Carlo simulation – the resulting computing effort would often be very significant.

Inspired by (20) and Boyle, Broadie & Glasserman (1997), we now introduce a direct technique to calculate the derivative that avoids some of the problems of the finite difference approach. Combining (7) and (11), we get:

$$\partial_{k,\xi} V = \int_{\mathbb{R}_+^N} f(\mathbf{T}) \frac{d}{d\mu} \varphi(\mathbf{T}; \boldsymbol{\lambda} + \mu \xi \mathbf{e}_k) \Big|_{\mu=0} d\mathbf{T} = \int_{\mathbb{R}_+^N} f(\mathbf{T}) \frac{d}{d\mu} \ln \varphi(\mathbf{T}; \boldsymbol{\lambda} + \mu \xi \mathbf{e}_k) \Big|_{\mu=0} \varphi(\mathbf{T}) d\mathbf{T}$$

that is, we can calculate the required derivative by, in effect, substituting for  $f$  a new payout function  $f^*(\boldsymbol{\tau}) = f(\boldsymbol{\tau}) \frac{d}{d\mu} \ln \varphi(\boldsymbol{\lambda} + \mu \xi \mathbf{e}_k) \Big|_{\mu=0} \equiv f(\boldsymbol{\tau}) h_{k,\xi}(\boldsymbol{\tau})$ . The required stochastic weights  $h_{k,\xi}$  can be calculated explicitly from (3):

$$h_{k,\xi}(\boldsymbol{\tau}) = \frac{\xi(\tau_k)}{\lambda_k(\tau_k)} - \int_0^{\tau_k} \xi(u) du + \frac{(v+1)v^{-1}Z_k}{1+v^{-1}Z_k^2} \alpha(\tau_k, Z_k) - \frac{(1+N/v)\alpha(\tau_k, Z_k)}{1+v^{-1}\mathbf{Z}^T \boldsymbol{\Sigma}^{-1} \mathbf{Z}} \mathbf{e}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{Z} \quad (24)$$

where, as before,  $Z_k = \tau_{1,v}^{-1}(p_k(\tau_k))$ , and:

$$\alpha(\tau_k, Z_k) \equiv \frac{1-p_k(\tau_k)}{\eta_{1,v}(Z_k)} \int_0^{\tau_k} \xi(u) du$$

We note that the Gateaux shifts we are often most interested in are of the form  $\xi_k(t) = 1_{t \in [T_1, T_2]}$  in which case the expressions above simplify considerably.

In terms of changes to the simulation algorithm, incorporating the derivative calculation above is simply a matter of inserting an additional calculation of the necessary derivative weights  $h_{k,\xi}$  after step v in our basic algorithm. The computation of  $\mathbf{Z}^T \boldsymbol{\Sigma}^{-1} \mathbf{Z}$  in (24) can be reduced to  $O(N)$  operations (as  $\mathbf{Z}^T \boldsymbol{\Sigma}^{-1} \mathbf{Z} = \mathbf{X}^T \mathbf{X}$ ), whereby the computational bottleneck of the delta-computations becomes the expression  $\mathbf{e}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{Z} = \sum_{i=1}^N Z_i \Sigma_{ik}^{-1}$ , which is of order  $O(N)$  per company  $k$ . That is, to calculate  $h_{k,\xi}$  for all  $k = 1, \dots, N$  will involve  $O(N^2)$  operations per path. We stress that the payout function needs only to be evaluated once per Monte Carlo loop, and the weights  $h_{k,\xi}$  are independent of the payout function. The latter point allows us to efficiently reuse the weights

when evaluating multiple payout functions at once (as is the case, say, in the simultaneous pricing of different tranches of a CDO).

The formula (24) holds whether or not the correlation matrix has a factor structure or not. As the  $O(N^2)$  operations can become quite significant for large  $N$ , we now wish to take advantage of the assumed correlation factor-structure to reduce the cost of the algorithm. As before, set  $\boldsymbol{\Omega} = (\mathbf{X}, g)$ . Conditioned on this vector,  $\mathbf{Z}$  is an  $N$ -dimensional vector of independent Gaussian samples with mean  $\mathbf{m} = \sqrt{v/g} \mathbf{cX}$  and standard deviation  $\mathbf{s} = (\sigma_1, \dots, \sigma_N)$ ,  $\sigma_i = \sqrt{v/g} \times \sqrt{F_{ii}}$ . We can then define a conditional Gateaux derivative as:

$$\partial_{k,\xi} V \Big|_{\boldsymbol{\Omega}} = \int_{\mathbb{R}_+^N} f(\mathbf{T}) \frac{d}{d\mu} \ln \varphi(\mathbf{T} | \boldsymbol{\Omega}; \boldsymbol{\lambda} + \mu \xi \mathbf{e}_k) \Big|_{\mu=0} \varphi(\mathbf{T} | \boldsymbol{\Omega}) d\mathbf{T} \equiv \int_{\mathbb{R}_+^N} f(\mathbf{T}) h_{k,\xi}^{\boldsymbol{\Omega}}(\mathbf{T}) \varphi(\mathbf{T} | \boldsymbol{\Omega}) d\mathbf{T}$$

where the Gaussian density  $\varphi(\mathbf{T} | \boldsymbol{\Omega})$  and the weights  $h_{k,\xi}^{\boldsymbol{\Omega}}(\mathbf{T})$  are easily shown to be:

$$\varphi(\mathbf{T} | \boldsymbol{\Omega}) = \prod_{k=1}^N \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2} \left(\frac{z_k - m_k}{\sigma_k}\right)^2\right) \lambda_k(T_k) (1 - p_k(T_k)) / \eta_{1,v}(z_k), \quad (25)$$

$$h_{k,\xi}^{\boldsymbol{\Omega}}(\mathbf{T}) \equiv \frac{\xi(T_k)}{\lambda_k(T_k)} - \int_0^{T_k} \xi(u) du + \frac{(v+1)v^{-1}z_k}{1+v^{-1}z_k^2} \alpha(T_k, z_k) - \frac{E_k}{\sigma_k} \alpha(T_k, z_k)$$

where  $z_k = \tau_{1,v}^{-1}(p_k(T_k))$  and where we assume that  $\sigma_k > 0$  for all  $k$ .

Importantly, by the law of iterated conditional expectations, simply applying our basic Monte Carlo algorithm to the modified payout  $f^*(\boldsymbol{\tau}) = f(\boldsymbol{\tau}) h_{k,\xi}^{\boldsymbol{\Omega}}(\boldsymbol{\tau})$  will give an unbiased estimate for the derivative  $\partial_{k,\xi} V$ . For given  $\mathbf{m} = \sqrt{v/g} \mathbf{cX}$  (which is calculated in step ii of the basic Monte Carlo algorithm) we note that (25) requires only  $O(N)$  operations for the joint estimation of  $\partial_{k,\xi} V$  for all  $k$ . This is, of course, a consequence of the conditional independence of  $\mathbf{Z}$  given  $\boldsymbol{\Omega}$ , which removes the term  $\boldsymbol{\Sigma}^{-1} \mathbf{Z}$  from (24). The reduction from  $O(N^2)$  to  $O(N)$  can, for large baskets, easily save a factor of two to three, with very minor changes to the basic algorithm.<sup>7</sup>

While not an application that would normally justify Monte Carlo simulations, for reference we now list some simulation results for (parallel shift) hazard rate sensitivity for a regular six-tranche synthetic CDO. Let  $v_{FD}$  and  $T_{FD}$  be the sample variance and total computation time associated with the finite difference method, and let  $v_{PW}$  and  $T_{PW}$  be defined analogously for the conditional payout weighting method in (25). The efficiency of the latter method relative to the former is then here defined as the ratio:

$$ER = T_{FD} v_{FD} / (T_{PW} v_{PW})$$

In the example shown in table C, the payout weighting method is many orders of magnitude faster than the finite difference method, even for the relatively straightforward payout function used. For small to medium CDOs, the sample variances of the former method are also considerably smaller than those of the latter, but this picture eventually reverses for large CDOs (for  $N = 150$ , the two methods produce about the same sample variances). Note that the finite difference numbers in the table are based on 1-basis-point shifts. Increasing the shift size will, as discussed earlier, increase the relative efficiency of this method, at the expense of a convexity bias in the resulting numbers. For the example above, we find empirically that an increase in the shift size to 5bp, say, improves the net efficiency of the finite difference results method by about a factor of three to four. On the other hand, increases in deal maturity, CDS spreads and deal complexity will tend to reduce the relative efficiency of the finite difference method, sometimes dramatically so. In practice, some experimentation is generally needed to

<sup>6</sup> Notice that if  $\Delta \times \xi$  is chosen to be negative, we know that  $\delta_{k,\Delta}$  will be zero for paths where  $\tau_k$  lies beyond the maturity of the basket derivative. For these paths, steps (vi)–(vii) should be skipped for performance reasons

<sup>7</sup> For the data in table C, the factor-structure efficiency savings vary from 10% ( $N = 25$ ) to 209% ( $N = 150$ )

<sup>8</sup> For instance, the finite difference technique could be applied to calculate low-spread deltas, and the payout weighting technique to medium- and high-spread deltas. See footnote 6 for the rationale behind this (the lower the spread, the less likely it is that a default will take place before the deal maturity)

### C. Timing and efficiency of hazard rate sensitivities of six-tranche CDO

#### A. Timing results (seconds). T: price only

| N   | T   | T <sub>PW</sub> | T <sub>FD</sub> | T <sub>PW</sub> /T <sub>FD</sub> |
|-----|-----|-----------------|-----------------|----------------------------------|
| 25  | 83  | 84              | 448             | 18.7%                            |
| 50  | 130 | 132             | 1,169           | 11.3%                            |
| 100 | 219 | 226             | 3,589           | 6.3%                             |
| 150 | 320 | 331             | 10,040          | 3.3%                             |

#### B. Efficiency ratios for hazard rate sensitivities for tranches A-F

| N   | ER <sub>A</sub> | ER <sub>B</sub> | ER <sub>C</sub> | ER <sub>D</sub> | ER <sub>E</sub> | ER <sub>F</sub> |
|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 25  | 124.1           | 84.6            | 79.1            | 145.5           | 111.0           | 253.1           |
| 50  | 41.8            | 44.4            | 45.4            | 70.8            | 55.6            | 117.0           |
| 100 | 13.3            | 30.1            | 23.1            | 38.5            | 32.4            | 64.1            |
| 150 | 15.6            | 19.3            | 20.6            | 31.2            | 24.9            | 52.0            |

Notes: all numbers based on  $m = 4 \times 10^5$  Student-*t* Monte Carlo simulations on a 1GHz PC. Homogeneous pool with all CDS spreads equal to 175bp and all copula correlations 25%, allowing for a one-factor correlation structure ( $M = 1$ ). The deal maturity is five years (quarterly schedule). Irrespective of the pool size  $N$ , the tranche definitions were as follows: A: [0%, 2%]; B: [2%, 5%]; C: [5%, 10%]; D: [10%, 15%]; E: [15%, 25%]; F: [25%, 100%]. For the finite difference sensitivities, a shift of 1bp was used along with the speed-up trick in footnote 6

optimise performance for a particular trade. In some cases, a 'mix-and-match' approach might lead to the best overall performance.<sup>8</sup>

While application of the techniques discussed above in many cases can help to improve the efficiency of Monte Carlo calculation of hedge parameters, the problem is still inherently slow and can take hours when run on a single CPU. For instance, for a 100-name STCDO, achieving merely 10% relative sample standard deviation on hazard rate sensitivities might require 10 million samples and nearly an hour of computation time on a 1GHz PC. A brute-force way to overcome this problem is, of course, to split the task out on multiple CPUs in a computer cluster. The 'embarrassingly' parallel nature of Monte Carlo makes this a straightforward exercise. Another, complementary, approach involves the application of variance reduction techniques to improve the convergence of the Monte Carlo algorithm itself. While application of variance reduction techniques to credit basket derivatives is outside the scope of this article, we point out that the high dimension ( $N$ ) of many typical basket derivatives renders many standard techniques (antithetic variates, stratified sampling, quasi-random sequences, etc) fairly useless. Application of likelihood ratio methods (see, for example, Glasserman, Heidelberger & Shahabuddin, 1999) appears to be a promising area of research, although again the high dimensions makes the determination of optimal likelihood ratios rather involved. For instruments with STCDO-like characteristics, an obvious – and often very effective – technique is to use the quasi-analytical formulas discussed earlier to set up one or more plain vanilla STCDOs as control variates.

#### Concluding remarks

This article has demonstrated a number of techniques to improve the efficiency with which prices and hedge parameters can be calculated for credit basket derivatives. For payouts depending on the basket loss, an extremely efficient quasi-analytical framework was introduced, which combined factor analysis with combinatorial methods. The factor analysis was backed up by efficient algorithms to factor-approximate arbitrary correlation matrices.

Using different combinatorics, one can extend the quasi-analytical argument in a number of ways, allowing for application to other securities than shown here (such as *n*th-to-default swaps) and to more realistic models, including those where the recovery is stochastic. Still, in the most general case, Monte Carlo methods are required. We have demonstrated a few useful techniques to perform hedge calculation in a Monte Carlo setting, but much work remains, particularly in design of efficient variance reduction techniques.

### Appendix. Choosing the loss unit

If all losses in default for individual names are integer multiples of the loss unit, there will be no approximation error arising from the discretisation of the loss distribution. We shall use the term *theoretical loss unit* for the largest value with this property. Since the computational effort is roughly inversely proportional to the size of the loss unit, the theoretical loss unit leads to the least possible computational effort without discretisation error. In some cases of practical interest, it is indeed possible to use the theoretical loss unit, but in general some approximation error must be accepted in order to reduce the computational effort.

For a given loss unit  $u$  the error for a loss  $W_k$  is given by  $|W_k - w_k u|$ , where  $w_k$  is the ratio  $W_k/u$  rounded to the nearest integer. This error (or rather its maximum over the basket) translates roughly into an error in the loss distribution and hence into an error in the apportioning of losses between tranches. Consequently the size of this error should be set at the level of pricing error tolerance for individual tranches. Note that sensitivities (to hazard rates, defaults, etc) will be much less affected by discretisation error.

Suppose that this tolerance is given and denote it  $\epsilon$ . If  $W$  is the smallest loss value ( $> \epsilon$ ), we define  $u_n := W/n$ ,  $n = 1, 2, \dots$ . We further define  $w_k^{(n)}$  as the integer rounded ratio  $W_k/u_n$ . Let  $\tilde{n}$  be the smallest integer<sup>1</sup> such that  $|W_k - w_k^{(n)} u_n| < \epsilon$  for all  $k$ . Then we take as our loss unit:

$$u \equiv \sum_{k=1}^N W_k / \sum_{k=1}^N w_k^{(\tilde{n})}$$

Note that the final step attempts to preserve the sum of all losses. Also note that this algorithm will produce the theoretical loss unit whenever this is greater than  $\epsilon$ . ■

<sup>1</sup> In practice, this is quickly found by a linear search beginning at  $n = 1$

Finally, let us note that while the ability to efficiently compute various risk measures is a prerequisite for good hedging, there is much left to be said about the application of the risk measures in actual hedging strategies with transaction costs and discrete-time re-balancing. As the gamma properties of CDOs are often quite complicated, such hedge analysis is an interesting area of future research with a number of open empirical and theoretical issues. ■

**Leif Andersen is a managing director, Jakob Sidenius is a principal and Susanta Basu is a vice-president at Banc of America Securities. They wish to express their gratitude to Jesper Andreasen and Vladimir Piterberg for constructive comments and ideas. E-mail: leif.andersen@bofasecurities.com, jakob.sidenius@bankofamerica.com and susanta.basu@bofasecurities.com**

#### REFERENCES

- Boyle P, M Broadie and P Glasserman, 1997**  
*Simulation methods for security pricing*  
Journal of Economic Dynamics and Control 21, pages 1,267–1,321
- Frey R and A McNeil, 2001**  
*Modelling dependent defaults*  
Working paper, University of Zurich
- Glasserman P, P Heidelberger and P Shahabuddin, 1999**  
*Asymptotically optimal importance sampling and stratification for pricing path-dependent options*  
Mathematical Finance 9, pages 117–152
- Gregory J and A Laurent, 2003**  
*I will survive*  
Risk June, pages 103–107
- Mashal R and M Naldi, 2002**  
*Extreme events and default baskets*  
Risk June, pages 119–122
- Merino S and M Nyfeler, 2002**  
*Calculating portfolio loss*  
Risk August, pages 82–86
- Pykhtin M and A Dev, 2003**  
*Coarse-grained CDOs*  
Risk January, pages 113–116
- Schmidt W and I Ward, 2002**  
*Pricing default baskets*  
Risk January 2002, pages 111–116