

# Credit Pricing Model Monte Carlo Acceleration

A new procedure is presented to accelerate the convergence of Monte Carlo simulations using the Default Correlation model. It is found that the modifications have been implemented correctly, and that the modifications result in a substantial improvement in the convergence rate of the Monte Carlo simulation models.

A particular benefit of the model is a dramatic improvement in the stability of delta computations, allowing more accurate and timely risk management of basket trades. The reason for the improvement in stability of the deltas is a change in the method of their calculation. The new method is found to be quite accurate for low and moderate correlations and is explained in detail in the next section.

The new method generates small negative delta amounts in certain cases. Based on the numerical results, these negative deltas are not confined to situations of high correlation and/or large spread differences between credits. The negative deltas do seem, however, to be consistent with zero and to therefore be the result of a noise component introduced in certain cases by the acceleration algorithm.

A typical Monte Carlo simulation algorithm assigns each simulation run, or path, an identical probability weight. If we allow, however, a different probability to be assigned to each simulation path, we allow more flexibility in the simulation and can thus shape the simulation in accordance with our needs. For example, we can choose the probabilities of each path in manner such that the simulation is guaranteed to reproduce the prices of “benchmark” securities, whose prices are known from market data (see <https://finpricing.com/curveVolList.html>). A simulation thus calibrated to benchmarks will then price off-market securities in a realistic manner.

The technique of assigning probability weights has, at least theoretically, the additional benefits of accelerating the convergence of the simulation, as well as allowing the sensitivities of the simulated price with respect to the benchmark securities to be computed without needing to perform additional simulations.

To see how this works, consider that a contingent claim has a value  $v_i$  for simulation path  $i$ . The value computed according to equally-weighted Monte Carlo will be given by

$$\langle V \rangle = \frac{1}{N} \sum_{i=1}^N v_i \quad (1)$$

whereas for non-uniformly weighted Monte Carlo we have

$$\langle V \rangle = \sum_{i=1}^N v_i p_i \quad (2)$$

where  $p_i$  is the probability weight for path  $i$ .

Obviously then, we require a method of computing the probability weights  $p_i$ . Let us consider that we have  $M$  benchmark instruments  $B_j$ . Let the value of the  $j$ -th benchmark instrument in the  $i$ -th simulation path be denoted  $w_{ij}$ . Then, in order to match the benchmark instruments, as required, we must have

$$B_j = \sum_{i=1}^N w_{ij} p_i \quad (3)$$

for all  $j$ .

Given that the number of benchmark instruments  $M \ll N$ , the number of simulations, the system of equations (3) is severely underdetermined, admitting possibly an infinite number of solutions. The question thus arises as to which of the many solutions to choose.

Intuitively, it makes sense to choose a solution which is as close as possible, in some sense, to the standard  $p_i = N^{-1}, \forall i$ , while simultaneously matching the benchmark prices. The task now is to define, in this context, what we mean by “as close as possible”. To do this, let

$$\Delta = \sum_{i=1}^N \Psi(p_i)$$

where  $\Psi(x)$  is a convex function. The search for a set of probabilities  $p_i$  which minimize  $\Delta$  subject to the constraints (3) can be performed by introducing  $M$  Lagrange multipliers  $\lambda_j$  and solve the Lagrangian dual problem, given by

$$\min_{\lambda} \max_p \left( \Delta + \sum_{j=1}^M \lambda_j \left( \sum_{i=1}^N g_{ij} p_i - B_j \right) \right) \quad (4)$$

There are many possible choices for  $\Psi(x)$ , among them

$$\Psi(x) = x \log x$$

which corresponds to the relative entropy,

$$\Psi(x) = \sqrt{x}$$

which corresponds to the square-root (Skorohod) distance, and

$$\Psi(x) = \left( x - \frac{1}{N} \right)^2$$

which corresponds to the least-squares distance.

The formal solution to the optimization problem (4) is given by

$$p_i = (\Psi')^{-1} \left( - \sum_{j=1}^M \lambda_j g_{ij} \right).$$

Here we will consider the relative entropy minimization,  $\Psi(x) = x \log x$ . It can be shown that the probability weights  $p_i$  which maximize (4) given a set  $\lambda_j$  is given by

$$p_i = \frac{1}{NZ(\lambda)} \exp \left( \sum_{j=1}^M g_{ij} \lambda_j \right) \quad (5)$$

where  $Z(\lambda)$  is the partition function, defined as

$$Z(\lambda) = \frac{1}{N} \sum_{i=1}^N \exp \left( \sum_{j=1}^M g_{ij} \lambda_j \right). \quad (6)$$

The minimum of (4) over  $\lambda_j$  can be obtained by minimizing the objective function

$$W(\lambda) = \log Z(\lambda) - \sum_{j=1}^M \lambda_j B_j, \quad (7)$$

then if  $\lambda_j^*$  are the Lagrange multipliers which minimize (7), we can compute the necessary path probabilities  $p_i^*$  as

$$P_i^* = \frac{1}{NZ(\lambda^*)} \exp\left(\sum_{j=1}^M \lambda_j^* g_{ij}\right). \quad (8)$$

A slight modification of the above procedure can be made if it is felt that an exact match to the benchmark prices is undesirable. This situation may occur due to liquidity considerations, or other difficulties in obtaining accurate prices for the benchmark securities. In this case, we allow a small discrepancy between the input benchmark prices and those implied by the calibrated simulation. This effectively means that we prefer calibrated path probabilities which are closer to the uniform  $p = N^{-1}$  in exchange for slight mismatches in the benchmarks. This shifts the relative importance of the benchmarks versus the relative entropy, placing more emphasis on entropy.

To allow this slight mismatch of benchmarks, we introduce a tolerance term into the objective function  $W(\lambda)$ , as

$$\tilde{W}(\lambda) = \log Z(\lambda) - \sum_{j=1}^M \lambda_j B_j + \frac{\varepsilon}{2} \sum_{j=1}^M \alpha_j \lambda_j^2 \quad (9)$$

where  $\varepsilon$  is an overall tolerance factor, while  $\alpha_i$  is an optional tolerance factor, which may in general be a function of the benchmark price  $B_j$ .

There exist many methods by which the objective function  $W(\lambda)$  or  $\tilde{W}(\lambda)$  may be minimized, some of which require explicit computation of the gradient of (7) or (9). This is given by

$$\frac{\partial W(\lambda)}{\partial \lambda_k} = \frac{1}{Z(\lambda)} \sum_{i=1}^N g_{ki} \exp\left(\sum_{j=1}^M \lambda_j g_{ij}\right) \quad (10)$$

and

$$\frac{\partial \tilde{W}(\lambda)}{\partial \lambda_k} = \frac{\partial W(\lambda)}{\partial \lambda_k} + \varepsilon \alpha_k \lambda_k. \quad (11)$$

The path weighting procedure lends itself to an efficient method of calculating sensitivities with respect to the input benchmark prices, or deltas. The major assumption which makes this application of the method practicable is that the distributions involved in the Monte Carlo simulation from which the paths are generated do not depend strongly on the benchmark prices. If this is the case, then we can reuse the simulation paths from one simulation run, re-calibrating them to match perturbed benchmark prices. The security in question is then repriced with the new path weights, enabling the computation of the hedge ratios.

Let the cashflows of a security in simulation path  $i$  be denoted  $G_i$ . We will then have

$$\langle G \rangle = \sum_{i=1}^N G_i p_i.$$

The sensitivity of  $\langle G \rangle$  with respect to the benchmark securities  $B_j$  is then written

$$\frac{\partial \langle G \rangle}{\partial B_j} = \sum_{k=1}^M \frac{\partial \langle G \rangle}{\partial \lambda_k} \frac{\partial \lambda_k}{\partial B_j}. \quad (16)$$

Using equation (5) a straightforward calculation shows that

$$\begin{aligned} \frac{\partial \langle G \rangle}{\partial \lambda_k} &= \sum_{i=1}^N G_i g_{ij} p_i - \left( \sum_{i=1}^N G_i p_i \right) \left( \sum_{i=1}^N g_{ij} p_i \right) \\ &= \langle G g_j \rangle - \langle G \rangle \langle g_j \rangle \\ &= \text{cov}(G, g_j) \end{aligned} \quad (17)$$

and since our benchmark securities are given by equation (3), then we have

$$\begin{aligned}\frac{\partial B_j}{\partial \lambda_k} &= \langle g_j g_k \rangle - \langle g_j \rangle \langle g_k \rangle \\ &= \text{cov}(g_j, g_k)\end{aligned}\quad (18)$$

from which we have the result that

$$\frac{\partial \langle G \rangle}{\partial B_j} = \sum_{k=1}^M [\text{cov}(G, g_k)]_k [\text{cov}(g_k, g_j)]_{kj}^{-1}.\quad (19)$$

If we perform a linear regression analysis of the  $G_i$  over the benchmark expectations  $g_j$ , that is we compute

$$\min_{\beta} \left[ \text{var} \left( \langle G \rangle - \beta_0 - \sum_{j=1}^M \beta_j g_j \right) \right]\quad (20)$$

then we find that

$$\beta_j = \frac{\partial \langle G \rangle}{\partial B_j}.\quad (21)$$

Of course, if the sensitivities desired are not with respect to the benchmark securities, then one has more work to do. In our case, however, equation (19) is exactly what we want.

It can be shown that in the case that the path weights  $p_i$  are computed using the relative entropy, that the regression (20) can be replaced by a re-computation of the weights using the simulation paths already computed, and the perturbed benchmark securities.

The constraints corresponding to equation (3) for the case of credit contingent contracts can be written in the form of default probabilities, since there is a one-to-one correspondence between

between the input credit spreads (prices) and the unconditional default probabilities of individual names.

We are thus able to compute, for each name involved in a given trade, the probability  $S(t)$  that the name will survive until time  $t$ . Let  $S_i(t_j)$  be the unconditional probability that name  $i$  survives until time  $t_j$ . Then we can write

$$S_i(t_j) = \sum_{k=1}^N g_{kij} p_k \quad (12)$$

where

$$g_{kij} = \begin{cases} 1 & \text{if } \tau_{ki} > t_j \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where in turn  $\tau_{ki}$  is the default time of name  $i$  in the  $k$ -th simulation path.

However, given the nature of the Monte Carlo simulation used for the pricing of credit derivatives, a more efficient implementation of the constraints (12) can be obtained if we instead define the default probability  $D_i(t_j)$  as

$$D_i(t_j) = S_i(t_{j-1}) - S_i(t_j)$$

which is the probability that  $t_{j-1} < \tau_i < t_j$ , ie., that name  $i$  defaults between times  $t_{j-1}$  and  $t_j$ . We then write

$$D_i(t_j) = \sum_{k=1}^N \tilde{g}_{kij} p_k \quad (14)$$



where  $\tilde{g}_{kij}$  is the “dual” of  $g_{kij}$ , that is

$$\tilde{g}_{kij} = \begin{cases} 1 & \text{if } t_{j-1} < \tau_{ki} < t_j \\ 0 & \text{otherwise} \end{cases}. \quad (15)$$

Running the simulation allows us to track  $\tilde{g}_{kij}$ , from which we can compute the partition function  $Z(\lambda)$ , and thus to determine those path weights which reproduce the input single-name default probabilities.