COMPARING DISCRETISATIONS OF THE LIBOR MARKET MODEL IN THE SPOT MEASURE

CHRISTOPHER BEVERIDGE, NICHOLAS DENSON, AND MARK JOSHI

Abstract. Various drift approximations for the displaced-diffusion LIBOR market model in the spot measure are compared. The advantages, disadvantages and implementation choices for each of predictor-corrector and the Glasserman–Zhao method are discussed. Numerical tests are carried out and we conclude that the predictor-corrector method is superior.

1. Introduction

It has become more common in recent years for insurance companies to invest their reserves in exotic structured products in order to obtain an increase in yield. Such products typically consist of a note issued by a bank which pays a coupon which is a path-dependent function of LIBOR or swap rates. This note is often callable, that is the bank can repay the principal early and terminate the contract. It is therefore important to be able to accurately assess the value of these products.

Methodologies for pricing such interest-rate products have evolved over time. The post Black–Scholes models for pricing interest rate derivatives were initially short rate models. These models relied on the notion of a hypothetical short rate chosen to follow a process that was typically some variant of a normal or log-normal process. One can view exotic options pricing as an extrapolation exercise, the objective is to extrapolate from the prices of vanilla derivative instruments such as swaps and bonds, and options thereon, to prices of exotic derivatives. One therefore had to fit the short rate process's parameters to the prices of market instruments. In particular, calibrating to both the discount curve and the prices of swaptions were non-trivial tasks. A consequence of all this fitting and the fact that everything was driven by a single short rate was that the dynamics of the model were not realistic. The interest rate discount curve can evolve in a complicated fashion which is not capturable by a single state variable.

The viewpoint later shifted to evolving market observable forward rates and in particular, the LIBOR market model has become popular. Its biggest virtue is the ability to easily calibrate to the discount curve and to caplet volatilities whilst still having enough flexibility to calibrate to other instruments and to specify correlations. Although it is now over ten years old, various details of its implementation have still not been fully resolved. In this paper, we focus on one of these: the best way to approximate the stochastic differential equation (SDE) evolution when using the discretely compounding money market account as numeraire.

Unlike in the Black–Scholes model, the stochastic differential equation describing the evolution of the forward rates involves a state-dependent drift. This means that there is no analytic solution to the SDE and it must be numerically approximated. A numerical approximation corresponds to a choice of discrete time model, we will compare two such approximations in this paper: the predictor-corrector approximation of Hunter, Jäckel and Joshi [10] and the "arbitrage-free" discretisation of Glasserman and Zhao [9]. Whilst there have been many other papers written on the topic of drift approximations in the LIBOR market model these focus on approximations when using the final bond as numeraire: [1], [7], [17], [15], [21], [22]. Here we are interested exclusively in the case where the numeraire is the discretely compounding money market account. Our motivation is that the variance of a Monte Carlo pricing simulation is generally lower when working with this numeraire, see for example [3]. No comparison between these two most popular methods appears to have been done previously.

This issue is of importance to practitioners as they want a Monte Carlo simulation which has low variance whilst suffering from only small discretisation errors. Whilst discretisation errors can always be reduced by taking extra steps, this will inevitably increase the time taken and it is therefore an undesirable solution.

We structure the paper as follows. In Section 2, the displaced diffusion version of the LIBOR Market Model is introduced, with corresponding discretisation schemes using both predictor corrector and Glasserman-Zhao methods. In Section 3, we present a number of issues with the most popular Glasserman-Zhao discretisations in any measure other than the terminal measure. Numerical results for forward rate agreements (FRAs), caplets, digital caplets, and target redemption notes (TARNs) are given in Section 4, and we conclude in Section 5.

2. Displaced-diffusion LIBOR Market Model

We recall the set-up of the LIBOR market model which is also often known as the BGM (Brace–Gatarek–Musiela) model or the BGM/J

(Jamshidian) model. For more detail, we refer the reader to the fundamental papers [4],[12], [18] and [20], or to the books [3], [5], [8], [14], [19] and [23]. The basic idea is to evolve discrete market-observable forward rates, rather than hidden unobservable factors. We have tenor dates $0 < T_0 < T_1 < \cdots < T_n$, with corresponding forward rates $f_0, \ldots f_{n-1}$, so that f_i is the forward rate for the period T_i to T_{i+1}. Set $\tau_i = T_{i+1} - T_i$.

Let $P(t, T)$ denote the price at time t of a zero-coupon bond paying one at its maturity, T. This paper primarily focuses on using the *spot LIBOR measure*, which corresponds to using the discretely-compounded money market account as numeraire, within the LIBOR market model. This numeraire is made up of an initial portfolio of one zero-coupon bond expiring at time T_0, with the proceeds received when each bond expires being reinvested in bonds expiring at the next tenor date, up until T_n. More formally, the value of the numeraire portfolio at time t will be,

$$N(t) = P(t, \eta(t)) \prod_{i=0}^{\eta(t)-1} (1 + \tau_i f_i(T_i)),$$

where $\eta(t)$ is the unique integer satisfying

$$T_{\eta(t)-1} \leq t < T_{\eta(t)},$$

and thus gives the index of the next forward rate to reset. The use of the discretely compounded money market account as numeraire within the LIBOR market model was introduced in [12]. When using this numeraire we will have to step to each forward rate reset time during a simulation because of the path dependent nature of the numeraire (see Remark 2 of [21]).

Under the displaced-diffusion LIBOR market model, the forward rates f_i are assumed to have the following evolution

$$df_i(t) = \mu_i(f, t)(f_i(t) + \alpha_i)dt + \sigma_i(t)(f_i(t) + \alpha_i)dW(t), \tag{2.1}$$

where the σ_i are deterministic F dimensional row vectors, the α_i are constant displacement coefficients, W is a standard F-dimensional Brownian motion under the relevant martingale measure, and the μ_i are determined by the no-arbitrage condition that all bond prices (i.e. model tradable assets) discounted by the numeraire must be martingales in the pricing measure. The σ_i are given by a pseudo square root (or suitable approximation) of the instantaneous covariance matrix for the f_i's.

Under the spot LIBOR measure, the drift term is given by

$$\mu_i(f,t) = \sum_{j=\eta(t)}^{n-1} \frac{(f_j(t)+\alpha_j)\tau_j}{1+f_j(t)\tau_j}\sigma_i(t)\sigma_j(t)',$$

see [5], where a $'$ is used to indicate the transpose of a matrix.

We consider displaced diffusion, as discussed in [14], [5], because it is a simple way to allow for the skews seen in implied caplet volatilities that have persisted in interest rate markets since the market events of 1998, see [14]. In any case, the results presented here will collapse to the non-displaced case when $\alpha_i = 0$ for all i.

In what follows, we will discuss evolution from time T_i to time T_{i+1}. Working with $\log(f_i + \alpha_i)$ terms is favorable as it yields state-independent volatility. We then have to evolve

$$d\log(f_j(t)+\alpha_j) = \left(\mu_j(f,t) - \frac{1}{2}\sigma_j(t)\sigma_j(t)'\right)dt + \sigma_j(t)dW(t),$$

with the drift term given above.

If drifts were not state-dependent, this equation would have an exact solution and simulation would be easy. Since they are state-dependent, approximations must be used, and as such, a large amount of work has been done on methods to approximate the term

$$\int_{T_i}^{T_{i+1}} \mu_k(f,t)dt, \tag{2.2}$$

that arises in simulation. The simplest approximation is Euler stepping, which uses

$$\hat{\mu}_k = \sum_{j=i+1}^{n-1} \frac{(f_j(T_i)+\alpha_j)\tau_j}{1+f_j(T_i)\tau_j}\tilde{\sigma}_k(T_i,T_{i+1})\tilde{\sigma}_j(T_i,T_{i+1})'.$$

to approximate (2.2), where $\tilde{\sigma}_k(T_i, T_{i+1})$ denotes any solution to the equation

$$\tilde{\sigma}_k(T_i,T_{i+1})\tilde{\sigma}_k(T_i,T_{i+1})' = \int_{T_i}^{T_{i+1}} \sigma_k(y)\sigma_k(y)'dy.$$

As is well documented in the literature, $\tilde{\sigma}_k(T_i, T_{i+1})$ is often used instead of $\sigma_k(T_i)$ to make the approximation of (2.2) more accurate, see [5]. We will use the $\tilde{\sigma}$ notation presented here throughout.

The Euler stepping method involves calculating the drift term assuming the f_j's are constant between $[T_i, T_{i+1}]$ and equal to their value at the start of the interval. More sophisticated approximations have been suggested in [15], [1], [21], and also ways to get around using such approximations, [9]. We will focus on two of the most popular methods, namely predictor-corrector [10], and the Glasserman-Zhao approach [9], presenting arguments against using the latter method when simulating under any measure other than the terminal measure.

2.1. **predictor corrector.** The predictor-corrector method was introduced to the LMM in [10] following Kloeden and Platen [16], and is sometimes referred to as the HJJ method after the authors. The idea is to evolve forward rates pretending all state variables are constant, recompute the drift at the evolved time and average the two drifts. We then re-evolve the forward rates using this averaged drift and the same random numbers. Formally, we use

$$\hat{\mu}_k = \frac{1}{2}\sum_{j=i+1}^{n-1}\left[\frac{(f_j(T_i)+\alpha_j)\tau_j}{1+f_j(T_i)\tau_j} + \frac{(\hat{f}_j(T_{i+1})+\alpha_j)\tau_j}{1+\hat{f}_j(T_{i+1})\tau_j}\right] \tilde{\sigma}_k(T_i,T_{i+1})\tilde{\sigma}_j(T_i,T_{i+1})',$$

to approximate (2.2), where a hat above a quantity implies that it has been estimated through an initial log-Euler evolution.

2.2. **Glasserman-Zhao.** The Glasserman-Zhao method instead uses a change of coordinates to simulate the forward rates. The idea is to map the forward rates to positive martingales. Being driftless and positive, we can use Euler-stepping to evolve the logs of these quantities preserving positivity and the martingale property. The forward rates can then be recovered from the evolved quantities. This method has the theoretical virtue that deflated bond prices can be made to be martingales even in the discretised model used for simulation. This is not the case for other discretisation methods such as predictor-corrector which provide a very good non-martingale approximation to a martingale measure. The major advantage is that the no-arbitrage condition described in [18] can be made to hold even in our discrete simulation, and as such, the prices of zero-coupon bonds and FRAs can be recovered without bias. On the negative side, while the Glasserman-Zhao method can price up to one caplet according to the Black formula, [2], without bias, there can be significant bias for other caplets, see [15] for the terminal case.

One can consider any number of possible changes of coordinates. However, here we will focus only on the most popular ones, which generally make up what is referred to as the Glasserman-Zhao method. In particular, we consider evolving the logarithm of the difference between adjacent deflated bonds, i.e. $\log((P(t,T_j) - P(t,T_{j+1}))/N(t))$, with minor adjustments to ensure positivity of discount ratios and to take account of displacements.

Define

$$D(t,T_j) = \frac{P(t,T_j)}{N(t)}, \tag{2.3}$$

$$= \prod_{k=0}^{j-1} \frac{1}{1+\tau_k f_k(t)}. \tag{2.4}$$

When $\alpha_j = 0$ for all j, we evolve the logs of

$$V_j(t) = (D(t,T_j) - D(t,T_{j+1})),$$
$$= \tau_j f_j(t) \prod_{k=0}^{j} \frac{1}{1+\tau_k f_k(t)},\ j = 0,\dots,n-1,$$

which is a positive martingale in continuous time. Euler stepping the logs of these V's in the simulation ensures each simulated V_j is a positive martingale and thus the recovered $D(.,T_j)$'s are monotonic decreasing in j, and are martingales. Note that the introduction of displacements means that in the continuous time specification of the LIBOR market model, forward rates can become negative, and thus discounted bond values need not be monotonic. As such, our discretisation should not try to force this by preserving strict monotonicity, but should instead aim to only allow non-monotonic discount ratios in line with that allowed in the continuous time specification. Thus, to introduce displacements, while maintaining a positive martingale with which to evolve our forward rates, we suggest using

$$V_j(t) = \tau_j(f_j(t) + \alpha_j) \prod_{k=0}^{j} \frac{1}{1+\tau_k f_k(t)},$$
$$= D(t,T_j) - (1-\tau_j\alpha_j)D(t,T_{j+1}),$$

which will remain positive provided $0 \le \alpha_j \le \frac{1}{\tau_j}$, as would be the case in any realistic set-up. Using such V's maintains all the properties that were present in the case of no displacements, except for the possibility of non-monotonic discount ratios, a shortcoming of the displaced diffusion LIBOR market model. This method of incorporating displacements in the Glasserman-Zhao method was originally suggested by Brace, [3].

Using the fact that the V's must be martingales in the pricing measure, we can derive their stochastic differential equations using the same methods as used in the original Glasserman-Zhao paper. We obtain,

$$\frac{dV_j(t)}{V_j(t)} = \left[\sigma_j(t) - \sum_{k=\eta(t)}^{j} \phi\left(\frac{V_k(t)}{D(t,T_k)}\right)\sigma_j(t)\right] dW(t), \tag{2.5}$$

where $\phi(x) = \min\{1, x^+\}$ is necessary to ensure the diffusion coefficient in (2.5) is Lipschitz-continuous in the discrete case. Equation (2.5) suggests a possible discretisation with

$$\hat{V}_j(T_{i+1}) = \hat{V}_j(T_i) \exp\left(-\frac{1}{2}\sigma_{\hat{V}_j}(T_i, T_{i+1})\sigma_{\hat{V}_j}(T_i, T_{i+1})' + \sigma_{\hat{V}_j}(T_i, T_{i+1})\tilde{Z}_i\right), \tag{2.6}$$

for $j = i+1, \ldots, n-1$, where

$$\sigma_{\hat{V}_j}(T_i, T_{i+1}) = \left[\tilde{\sigma_j}(T_i, T_{i+1}) - \sum_{k=i+1}^{j} \phi\left(\frac{\hat{V}_k(T_i)}{\hat{D}(T_i, T_k)}\right) \tilde{\sigma_k}(T_i, T_{i+1})\right], \tag{2.7}$$

where $\tilde{Z}_0, \tilde{Z}_1, \ldots$ denote independent F-dimensional standard normal column vectors, and hats are used to indicate simulated quantities. Note that under such a discretisation, it is easy to see that $E_i\left[\hat{V}_j(T_{i+1})\right] = \hat{V}_j(T_i)$, where $E_i(.)$ denotes the expectation conditional on the information available at time T_i in the discretised measure; so the simulated V's will remain positive martingales in the discrete model.

So far we have changed coordinates from forward rates to V_j's and have simulated a set of V_j's. All that we need to do now is recover the forward rates to price our products. We do this by recovering the discounted bond prices, which can easily be used to recover the forward rates, from

$$\hat{D}(T_{i+1}, T_j) = \frac{\hat{D}(T_i, T_{i+1}) - \sum_{k=i+1}^{j-1} \hat{V}_k(T_{i+1}) \prod_{l=k+1}^{j-1}(1 - \alpha_l \tau_l)}{\prod_{k=i+1}^{j-1}(1 - \alpha_k \tau_k)}, \tag{2.8}$$

for $j = i+2, \ldots, n$, with the understanding that $\sum_{k=i}^{j} = 0$ and $\prod_{k=i}^{j} = 1$ when $i > j$. Note that under the spot LIBOR measure, $D(T_i, T_{i+1}) = D(T_{i+1}, T_{i+1})$ since $D(t, T_{i+1})$ only depends on forward rates which have all reset by time T_i.

Since the $\hat{D}$'s are recovered as a linear combination of the $\hat{V}$'s, they will also remain martingales in the discrete simulation. However, as discussed below, using this exact change of coordinates allows the possibility of negative discount ratios.

It is important to realize that negative discount ratios are a much more serious issue that negative interest rates. As long as one does not allow the ability to hold cash in the economy, a negative interest rate does not imply an internal arbitrage in the model. However, a negative value for a zero-coupon bond implies that one can buy something for a negative amount that will have a guaranteed value of 1 in the future; this is clearly an arbitrage internal to the model.

3. Assessing the Glasserman-Zhao discretisation

In this section, we look at the various methods used to overcome the problem of negative discount ratios. We consider two interpretations of the method presented in [9] identifying significant short comings in each. We also consider an obvious way one could attempt to deal with these shortcomings, showing that it too is flawed. While we conduct our discussion in the spot LIBOR measure, it is important to realise that all the problems identified will naturally extend to measures based on hybrid numeraires, i.e. where some zero-coupon bond with expiry on one of the inner tenor dates is used as numeraire.

3.1. **The terminal measure.** Although our focus in on the spot measure, it is worth nothing why the issues with negative discount ratios do not arise in the *terminal measure.* For details of the discretisation in the terminal measure see [9].

Under the terminal measure the expression for recovering the deflated bond prices is given by

$$\begin{aligned}\hat{D}_j^{\text{terminal}} &= \prod_{k=j}^{n-1}(1-\alpha_k\tau_k) + \sum_{k=j}^{n-1}\hat{V}_k^{\text{terminal}}\prod_{l=j}^{k-1}(1-\alpha_l\tau_l),\\ &= 1+\sum_{k=j}^{n-1}\hat{V}_k^{\text{terminal}}, \text{ where } \alpha_i = 0 \;\forall\; i.\end{aligned}$$

Starting from $D_n \equiv 1$ we work backwards, each time adding another positive term to the previous $\hat{D}$ until $\hat{D}(T_{i+1}, T_{i+1})$ is reached. Therefore the $\hat{D}(., T_j)$'s are positive, as well as decreasing in j when displacements are zero so all forward rates will be positive.

3.2. **Method 1 - negative discount ratios.** As noted in [9], using (2.8) to recover the $\hat{D}$ terms from the $\hat{V}$'s can lead to negative $\hat{D}$'s. The reason this problem arises in the spot LIBOR measure, but not when using an equivalent simulation scheme in the terminal measure is that we are simulating the differences

$$\hat{V}_j = \hat{D}(., T_j) - (1-\alpha_j\tau_j)\hat{D}(., T_{j+1}),$$

but when recovering the $\hat{D}$'s we now start from $\hat{D}(T_i, T_{i+1})$ and work forward, calculating

$$\hat{D}(., T_{j+1}) = (\hat{D}(., T_j) - \hat{V}_j)/(1 - \alpha_j \tau_j)$$

from simulated and already-calculated quantities. This means the $\hat{D}$s are recovered as a difference; since there are no appropriate constraints, this can and does lead to negative value for $\hat{D}_j$.

3.3. **Method 2 - backwards method 1.** The problem of the previous method stemmed from working forwards in j, and thus requiring the recovered $\hat{D}$'s be calculated as a difference. An obvious alternative is therefore to find a way in which we can recover the $\hat{D}$'s by working backwards, starting from $\hat{D}(T_{i+1}, T_n)$. Glasserman-Zhao suggest adding a new term to the simulated quantities,

$$V_n = D_n,$$

with dV_n given by (2.5) where $\sigma_n(t) = 0$. Trivially, we can also extend (2.6) and (2.7) to hold for $j = n$ by imposing $\sigma_n(t) = 0$. Using $\hat{V}_n$, we can now recover the discounted bond prices by working backwards, giving

$$\hat{D}(T_{i+1}, T_j) = \sum_{k=j}^{n} \hat{V}_k(T_{i+1}) \prod_{l=j}^{k-1} (1 - \alpha_l \tau_l) \tag{3.1}$$

$$= \sum_{k=j}^{n} \hat{V}_k(T_{i+1}), \text{ where } \alpha_i = 0 \ \forall \ i. \tag{3.2}$$

This method ensures the $\hat{D}$'s remain positive martingales as the $\hat{V}$'s are positive martingales by construction.

There are two possible ways to use Glasserman-Zhao's idea. In this section, we focus on the seemingly theoretically correct interpretation, leaving the practically appealing implementation to the next section. The method discussed here is the one described in [3].

Within the new set-up, $\hat{D}(T_i, T_{i+1}) = \hat{D}(T_{i+1}, T_{i+1})$ is known at the start of the step from T_i to T_{i+1}, and will not change. Thus to recover the unknown $\hat{D}$'s, using (3.1), we need to simulate $\hat{V}_{i+2}, \ldots, \hat{V}_n$. Importantly, we are now simulating $\hat{V}_j$ for $j = i + 2, \ldots, n$ instead of for $j = i+1, \ldots, n-1$. The problem with this is that the properties of f_{i+1} enter the simulation only in an indirect and minimal way. To see this, consider (2.6) and (2.7). The term which does all the work in the simulation is (2.7), in which $\tilde{\sigma}_j$ dominates. As $\hat{V}_{i+1}$ is not simulated in this method, the volatility of the first forward rate not yet reset, σ_{i+1}, does not enter the discretisation in a complete way; it only has a partial influence through the summation term of (2.7) where all coefficients are less than or equal to one. If, for example, we are pricing a product that is highly sensitive to the first forward rate, such as a TARN with an increasing initial term structure of forward rates, we will not be simulating the important quantities accurately, and are likely to badly mis-price.

In addition, without displacements, there is nothing to ensure that $\hat{D}(T_{i+1}, T_{i+2}) < \hat{D}(T_{i+1}, T_{i+1})$. Since we obtain

$$\hat{f}_{i+1}(T_{i+1}) = \frac{\hat{D}(T_{i+1}, T_{i+1})/\hat{D}(T_{i+1}, T_{i+2}) - 1}{\tau_{i+1}},$$

negative forward rates can therefore be obtained which is an undesirable feature when displacement is zero.

In terms of the set-up used in [9], where forward rates are recovered directly from the $\hat{V}$'s, this method can equivalently be seen as using the relation,

$$\hat{D}(T_i, T_{i+1}) = \hat{D}(T_{i+1}, T_{i+1}) = \sum_{k=i+1}^{n} \hat{V}_k(T_{i+1}) \prod_{l=j}^{k-1} (1 - \alpha_l \tau_l), \tag{3.3}$$

to determine $\hat{V}_{i+1}(T_{i+1})$, which can then be used to recover the forward rates with the other $\hat{V}$'s. This ensures (3.3) holds.

3.4. **Method 3 - backwards method 2.** In order to address the problems highlighted in the previous section, a second more practically appealing interpretation of Glasserman-Zhao's idea can be used. Rather than worry about discount ratios that have already reset (i.e. that depend only on forward rates which have already reset), we can simulate $\hat{V}_j$ for $j = i+1, \ldots, n$ and recover $\hat{D}(T_{i+1}, T_j)$ for $j = i+1, \ldots, n$ using (3.1). This ensures the properties of each relevant forward rate fully enter the simulation, and that without displacements the relevant $\hat{D}$'s are monotonic, ensuring all recovered forward rates are positive.

An obvious disadvantage with this method is that we are required to simulate $n+1$ quantities in a model with an $n-$dimensional state space. This raises other issues. In particular, we are now recalculating $\hat{D}(T_{i+1}, T_{i+1})$ at time T_{i+1}, after it has already reset, and there is nothing to ensure that it will be equal to its value at the start of the step. For example, consider evolving from 0 to T_0. By definition, $D(T_0, T_0) = 1$. However, under this method we recalculate $\hat{D}(T_0, T_0)$ at time T_0, but do not ensure that it is equal to 1. A similar effect will occur at each evolution time. As such, additional discretisation bias has been introduced into the method.

On an even more practical level, the additional discretisation bias can cause problems when incorporating this method in pre-existing libraries. In particular, if the determination of payoffs and subsequent discounting are done independently, it is likely $N(T_{i+1})$ will be valued inconsistently with the pay-off, meaning the pricing of FRAs will not be bias-free. This follows since at each time T_{i+1}, the principal in the numeraire portfolio is given by

$$\prod_{j=0}^{i}(1+f_j(T_j)\tau_j) \equiv \frac{1}{\hat{D}(T_i,T_{i+1})},$$

which is naturally updated at the end of the previous step from T_{i-1} to T_i (i.e. without taking into account the additional bias described above). To see the effect of this, consider pricing the FRA from T_{i+1} to T_{i+2} struck at K within a larger simulation. We evolve to T_{i+1} and calculate the payoff as

$$\left[\frac{\hat{D}(T_{i+1},T_{i+1})/\hat{D}(T_{i+1},T_{i+2})-1}{\tau_{i+1}}-K\right]\frac{\hat{D}(T_{i+1},T_{i+2})}{\hat{D}(T_{i+1},T_{i+1})}.$$

We then divide by the numeraire value, $N(T_{i+1})$. If

$$N(T_{i+1}) = \frac{1}{\hat{D}(T_{i+1},T_{i+1})}, \tag{3.4}$$

the discounted payoff will be

$$\frac{\hat{D}(T_{i+1},T_{i+1})-\hat{D}(T_{i+1},T_{i+2})(1+K\tau_{i+1})}{\tau_{i+1}},$$

and we have no problems since this is a linear combination of martingales in the discrete model, and is therefore also a martingale. However, since additional bias has been introduced to $\hat{D}(T_{i+1},T_{i+1})$, (3.4) does not hold, and our FRAs are not priced without bias as the discounted payoff will no longer be a martingale (but instead the ratio of linear combinations of martingales) in the discrete simulation. More fundamentally, our discrete simulation is no longer arbitrage free. An example of a library where this would be an issue is *QuantLib*.

3.5. **Method 4 - re-normalisation.** To correct for the additional discretisation bias introduced by the previous method, an obvious solution is to use renormalization. In particular, to ensure that $\hat{D}(T_{i+1},T_{i+1}) =$

$\hat{D}(T_i, T_{i+1})$, we can rescale the $\hat{V}$'s simulated using (2.6) and (2.7). We replace $\hat{V}_j$ with

$$\bar{V}_j(T_{i+1}) = \frac{\bar{D}(T_i, T_{i+1})}{\sum_{k=i+1}^{n} \hat{V}_k(T_{i+1}) \prod_{l=j}^{k-1} (1 - \alpha_l \tau_l)} \hat{V}_j(T_{i+1}), \tag{3.5}$$

where the $\bar{D}$'s are recovered as in (3.1), but with the $\bar{V}$'s instead of the $\hat{V}$'s.

This will guarantee that discount ratios do not change after they have reset, while maintaining the practical advantages of the method presented in the previous section. However, the $\bar{V}$'s are now recovered as a ratio of linear combinations of the $\hat{V}$'s, and thus so are the $\bar{D}$'s. Since the $\hat{V}$'s are martingales by construction, the recovered $\bar{V}$'s and $\bar{D}$'s will no longer be martingales in the discretised model. While this re-normalisation has given desirable properties, it has destroyed the reason for using Glasserman-Zhao in the first place, which was to evolve forward rates whilst ensuring that discount ratios were martingales.

4. Numerical Results

To compare prices obtained using log-Euler, predictor-corrector, and the four Glasserman-Zhao methods we price FRAs, caplets, digital caplets, and TARNs. As a term structure we take yearly forward rates $f_0, f_1, \ldots, f_{19}$ (with the first reset at $t_0 = 1$), all equal to 5%. The instantaneous correlation between rate i and rate j is given by

$$\rho_{ij} = e^{-\beta|T_i - T_j|},$$

with $\beta = 0.04$. High correlation (small β) equates to larger drift terms and is therefore a tough test of the different methods. The instantaneous volatility is flat at 15% and the displacement for each forward rate is 1.5%.

We ran 2^{21} paths using Sobol quasi-random numbers with Brownian Bridging (for details of Brownian Bridging see [11]). This large number of paths was chosen to ensure that any errors are mostly due to the method rather than convergence.

Figure 4.1 shows the pricing errors for an at-the-money (ATM) FRA using log-Euler (LE), predictor-corrector (PC), and the four Glasserman-Zhao methods (GZ1 = Glasserman-Zhao method 1, etc.). The 20 points on the x-axis represent FRAs on the corresponding forward rates. The y-axis scale ranges from minus one basis point (bp) to 1bp. As we expect, the log-Euler method does not price FRAs without error. The two most interesting features are that predictor-corrector prices FRAs without any noticeable error and that the re-normalised Glasserman-Zhao

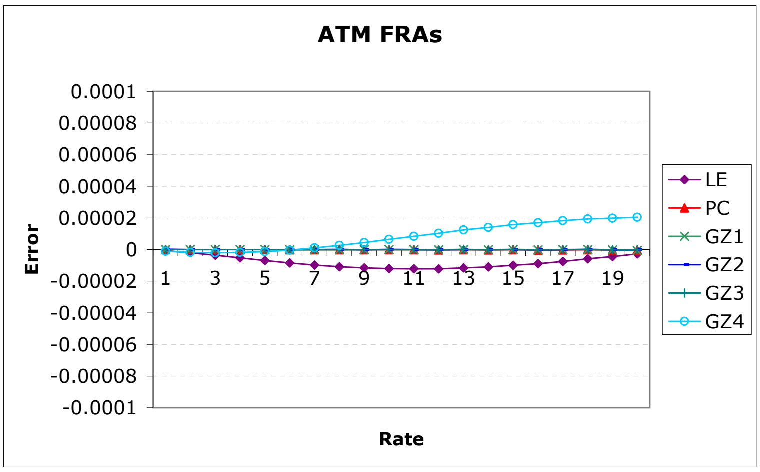

FIGURE 4.1. Pricing errors for at-the-money FRAs, strike = 0.05

method performs similarly to log-Euler. The error in the re-normalised Glasserman-Zhao method can be attributed to the destruction of the martingale property.

To compare caplet prices, we consider the difference between the Black price and the simulated prices for both ATM, figure 4.2 and out-of-the-money (OTM) caplets, figure 4.3. The prices of the ATM caplets rise from 35 basis points to 71 basis points and then fall back to 61 basis points. The errors are therefore on the order of 0.3 percent in relative terms.

The first three OTM caplet prices are $1.95E-5, 2.1E-4$ and $5.3e-4$ and the final one is worth 33 basis points. The errors are therefore very high in relative terms at the short end. Even at the long end, an error of one basis point is roughly 3 percent.

Similarly with FRAs, predictor-corrector prices both ATM and OTM caplets without any significant error. For the ATM caplets, Glasserman-Zhao methods 1 and 4 have errors of similar magnitude to log-Euler. The best performing Glasserman-Zhao method for ATM caplets is method 3, which involves simulating $n+1$ quantities. This suggests that the additional bias introduced by simulating already reset rates is small.

Focusing on OTM caplets, it appears that of the four Glasserman-Zhao methods, method 3 again prices with the smallest error. All four methods, however, perform significantly worse than predictor-corrector and arguably worse than log-Euler.

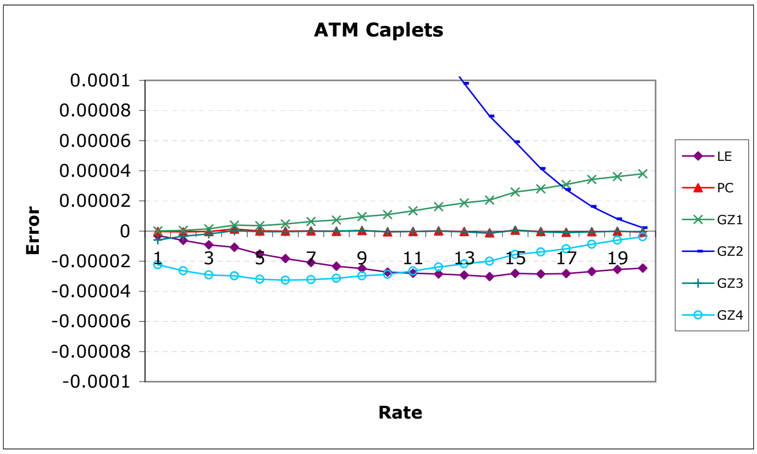

FIGURE 4.2. Pricing errors for at-the-money caplets

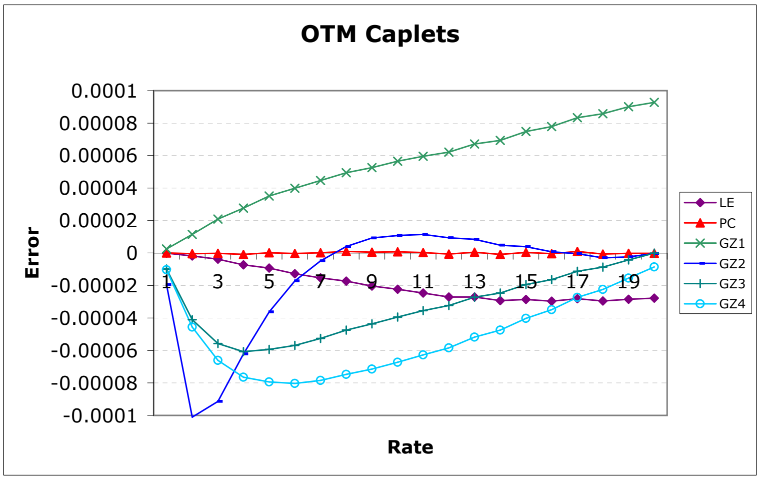

FIGURE 4.3. Pricing errors for out-of-the-money caplets, strike = 0.08

We now turn our attention to digital caplets. Figure 4.4 shows the bias for ATM digital caplets, where the scale ranges from -100bps to 100bps. Both predictor-corrector and log-Euler have only small biases, but the four Glasserman-Zhao methods are very significant biased. The prices of these digital caplets range from 0.4264 (i.e. 4264 basis points) to 0.1323. So an error of 100 basis points is on the order of 2.5 percent for the short-dated caplets.

This bias is also present in OTM digital caplets, where once again log-Euler and predictor-corrector perform better than the four Glasserman-Zhao methods. Products with digital effects provide a stronger test of

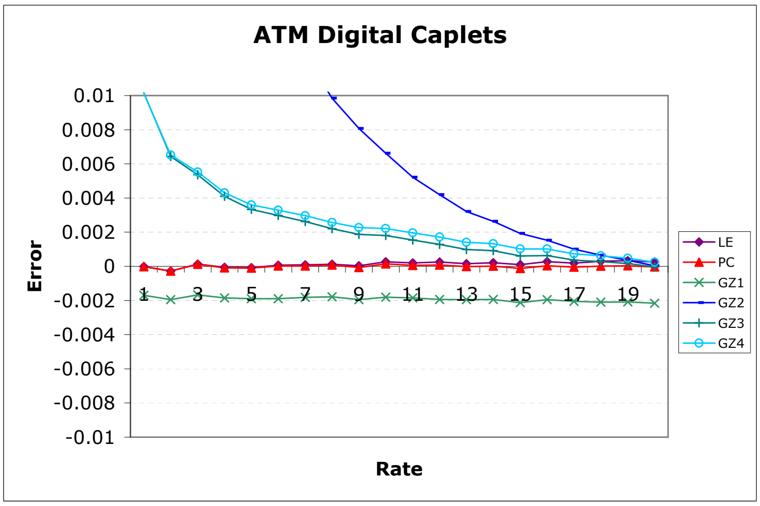

FIGURE 4.4. Pricing errors for at-the-money digital caplets

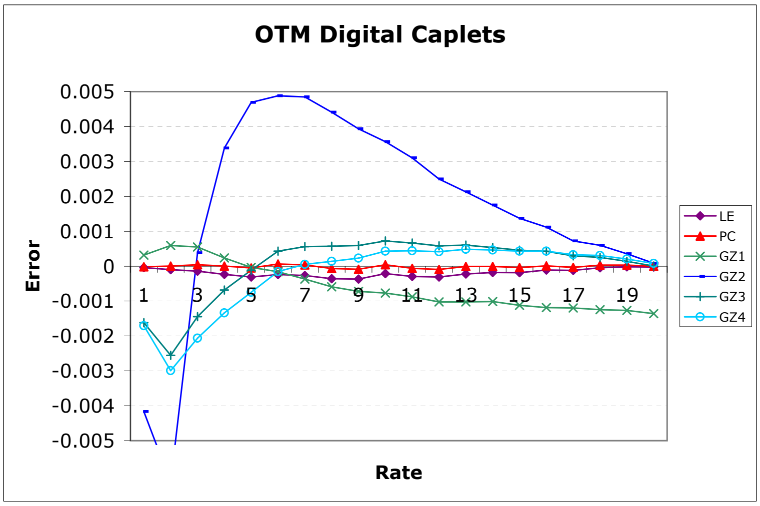

FIGURE 4.5. Pricing errors for out-of-the-money digital caplets, strike = 0.08

the approximation methods, since the effect of any bias is magnified significantly.

The prices of the first three OTM digital caplets are 41, 251 and 460 basis points, they then increase to 876 basis points before decreasing to 660 basis points. The errors displayed in Figure 4.5 are therefore significant in relative terms.

The pricing of digital caplets is an example of this, and the errors suggest that the Glasserman-Zhao methods will not price more exotic

Method	Time
LE	1.0
PC	1.3
GZ1	1.2
GZ2	1.2
GZ3	1.2
GZ4	1.2

TABLE 4.1. Timings from caplet prices for the various methods relative to the time taken for the fastest method: log-Euler.

products with digital type effects accurately, as we will see later with TARNs. The prices of these digital caplets range from

It is also worth investigating the time taken by each method. Table 4.1 shows that, for our implementations, predictor-corrector is slower than all the Glasserman-Zhao methods, however the difference is small.

Despite the fact that pricing FRAs, caplets, and digital caplets are good tests, we are ultimately more interested in how the different methods perform for popular exotic products which they are regularly used to price, such as TARNs (for a detailed description of a TARN see [6]). As there is no analytic price for a TARN, we plot prices against step size to see which method converges fastest.

We use a similar setup as before, but with an increasing term structure of forward rates starting from $t_0 = 0$. We have $f_0 = 2\%$ and $f_{20} = 10\%$, with forward rate f_i satisfying, $f_i = \log(a + b \times T_i)$ where the constants a and b are chosen to keep f_0 and f_{20} at their given values.

Figure 4.6 plots the convergence of prices. The TARN pays an inverse floating coupon, $c_i = \max(K - 2f_i, 0)$ with strike, $K = 0.08$ and a total coupon of 0.1. The principal is repaid either at maturity or when the total coupon is reached if that is sooner. The maturity is 20 years. The TARN returns the principal as soon the total coupon is reached. Since the coupons are inverse floating, this means that for an increasing yield curve either the principal is paid back early, or if rates go up then it goes to the full maturity paying no interest. This results in a large change in value according to whether a critical rate is above or below a threshold. It is therefore effectively a digital product.

Similarly to the previous examples predictor-corrector performs very well, converging the fastest. Log-Euler and Glasserman-Zhao method 1 also converge quickly, however, the three other Glasserman-Zhao methods do not perform well at all, which is consistent with the digital caplet examples.

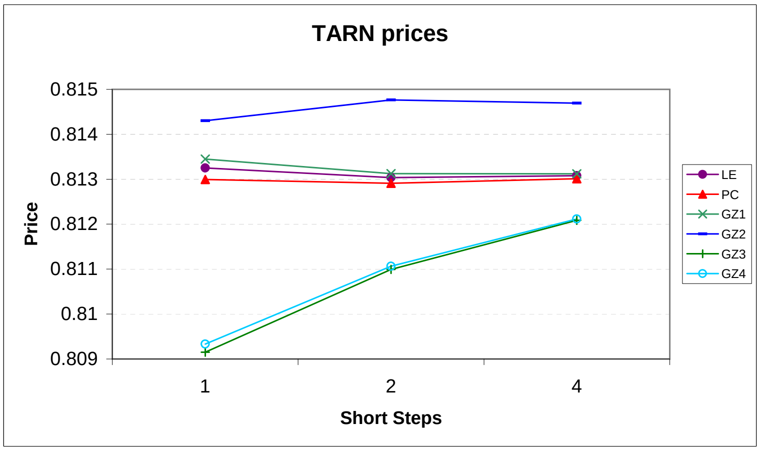

FIGURE 4.6. The price of a TARN varying the number of short steps per step for the six different drift approximation methods.

5. Conclusion

In conclusion, although the Glasserman-Zhao method appears to have the theoretical appeal of being an arbitrage-free discretization, we have seen that this is not really the case when working in the spot measure. In practical terms, our numerical results show that in this case predictor-corrector is more effective and we recommend its use, especially for products with a digital effect.

References

[1] P. Balland, Stoch-Vol for Libor Model, presentation at *International Centre for Business Information* (ICBI) conference, Madrid, 2004.
[2] F. Black, The pricing of commodity contracts, *Journal of Financial Economics* **3**, 167-179, 1976
[3] A. Brace, *Engineering BGM*, Chapman and Hall, 2007.
[4] A. Brace, D. Gatarek, M. Musiela, The market model of interest-rate dynamics, *Mathematical Finance* **7**, 127–155, 1997
[5] D. Brigo, F. Mercurio, *Interest Rate Models – Theory and Practice*, Springer Verlag, 2001
[6] C. Fries, M. Joshi, Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks, to appear in *Journal of Computational Finance.*
[7] A. Daniluk, D. Gatarek, A fully lognormal LIBOR market model, *Risk magazine*, September 2005, 115–118
[8] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
[9] P. Glasserman and X. Zhao, Arbitrage-free discretization of lognormal forward Libor and swap rate models, *Finance and Stochastics* **4** 35–68, 2000.
[10] C. Hunter, P. Jäckel and M. Joshi, Getting the drift, *Risk*, **14**(7) 81–84, 2001.

[11] P. Jäckel *Monte Carlo Methods in Finance*, Wiley, 2002.
[12] F. Jamshidian, LIBOR and swap market models and measures, *Finance and Stochastics* **1**, 293–330, 1997
[13] I. Karatzas, E. Shreve, *Brownian Motion and Stochastic Calculus*, Second edition, Springer Verlag, 1997
[14] M. Joshi, *The Concepts and Practice of Mathematical Finance*, Cambridge University Press, 2003.
[15] M. Joshi, A. Stacey, New and robust drift approximations for the LIBOR market model, to appear in *Quantitative Finance*
[16] P. Kloeden, E. Platen, *Numerical Solution of Stochastic Differential Equations,* Springer 1992
[17] O. Kurbanmuradov, K. Sabelfeld, J. Schoenmakers, Lognormal approximations to LIBOR market models, *Journal of Computational Finance,* Volume 10, Number 1, 2002
[18] M. Musisela, M. Rutkowski, Continuous-time term structure models: forward measure approach. *Finance and Stochastics,* **1** 261–292, 1997
[19] M. Musiela, M. Rutkowski, *Martingale Methods in Financial Modelling*, Springer Verlag, 1997.
[20] K. R. Miltersen, K. Sandmann, D. Sondermann, Closed form solutions for term structure derivatives with log-normal interest rates, *Journal of Finance,* **52**, (1) 409–430, 1997
[21] R. Pietersz, A. Pelsser, M. van Regenmortel (2004). Fast drift-approximated pricing in the BGM model, *Journal of Computational Finance,* Volume 8, Number 1, 2004
[22] R. Pietersz, A. Pelsser, M. van Regenmortel, Bridging Brownian LIBOR, *Wilmott Magazine,* 18, 98–103, 2005
[23] R. Rebonato, *The Modern Approach to Interest Rate Derivative Pricing*, Princeton University Press, 2002

Centre for Actuarial Studies, Department of Economics, University of Melbourne VIC3010, Australia
E-mail address: c.beveridge2@pgrad.unimelb.edu.au

E-mail address: ndenson@gmail.com

E-mail address: mark.joshi@unimelb.edu.au