

Computation of the Endogenous Mortgage Rates with Randomized Quasi-Monte Carlo Simulations

Yevgeny Goncharov, Giray Ökten, Manan Shah

Department of Mathematics

Florida State University

Tallahassee FL 32306

October 11, 2006

Abstract

The problem of computing the mortgage rate implied by a prepayment and interest rate model is considered. A Monte Carlo algorithm that uses a correlated sampling approach is introduced to simulate the model. Numerical results are used to compare Monte Carlo and randomized quasi-Monte Carlo methods with a numerical PDE solution. A particular randomized quasi-Monte Carlo method, random-start scrambled Halton sequence, gives superior performance, especially in high dimensions.

1 Introduction

A fixed-rate level-payment mortgage at first glance looks like a fixed-coupon bond. However, the cash flow of this mortgage is not certain, as its name may imply. Borrowers might prepay mortgages due to various reasons (in this case the lender “loses” anticipated interest payments¹). This uncertainty (prepayment option) makes the mortgage-based securities riskier than traditional fixed-coupon bonds.

Prepayment models try to predict the borrowers’ prepayment behavior based on information available to the market. There is still no agreement on the “correct” prepayment model. The major problem in prepayment modeling is that this is essentially a behavioral science. Currently, numerous attempts ([4], [11], [20]) to apply (often straightforwardly) the option-theoretic methodology to predict the exercise

¹This might be bad for the mortgage investor (if those payments are higher than the current interest rates can provide) or good (if the situation is the opposite).

of prepayment options did not produce satisfactory results. Consequently, the major trend on Wall Street (and in academia to some extent) is to model the prepayment behavior empirically. The prepayment rates are regressed against some explanatory variables. The 10-year Treasury yield is often considered as one of the most influential predictors.

If we ask “why the 10-year Treasury?” then the most common answer would be that this rate closely tracks the 30-year fixed-rate mortgage rate. So, in fact, the empirical models use some benchmark for the mortgage rate. This idea was implemented in the mortgage-rate based (or MRB for short, see [6]) approach to prepayment modeling. It assumes that the borrowers’ refinancing decision is based mainly on the comparison of the *contract* and *current* (available for refinancing) mortgage rates (the internet is abundant with “calculators” which tell one how much one “saves” if he/she refinances). If this approach to model the refinancing incentive is taken, then the investor needs to model the *mortgage rate* to specify the prepayment process which would then enable pricing or hedging mortgage-backed securities. This mortgage rate model needs to be in agreement with the mortgage rates implied by the resulting prepayment behavior and the underlying interest rate model (we call such mortgage rate models *endogenous*). The problem of finding the endogenous mortgage rate is known to be addressed in proprietary Wall Street systems, but not (widely) published. In spite of its importance, the problem was not considered in depth in academia until recently.

The empirical approach (currently widely used by practitioners) can be viewed as an MRB approach with an exogenously specified mortgage rate model. For example, a popular (in academia as well as in industry) choice would be that the mortgage rate is the 10-year Treasury yield plus some (exogenously specified) constant. The natural question, then, is whether the empirical models would perform better if we used the endogenous mortgage rate instead of some long interest rate predictor.

The first general endogenous mortgage rate model under a sub-optimal prepayment assumption was developed by Goncharov [6], where the endogenous mortgage rate is formulated as a fixed-point of a functional operator. Pliska [18] investigated the problem in a discrete setting.

In this paper, we consider this problem of computing the mortgage rate process under the MRB prepayment assumption. Since the

problem potentially is multi-factor (the interest rate, house price process, etc.), the PDE approach used in Goncharov [6] for a single factor implementation will not be computationally efficient in general. To overcome the curse of dimensionality, one might want to use Monte Carlo simulation. Additionally, the Monte Carlo method has a very important advantage for practitioners in the sense that one could very easily implement different kinds of interest rate models (e.g., jump-diffusion). However, Monte Carlo convergence is usually too slow for a real time analysis. Another problem is that we solve a fixed-point functional equation for mortgage rates and the application of simulation is not trivial in this case. In this paper, we will use randomized quasi-Monte Carlo (RQMC) methods to solve the model. We will consider a single factor implementation where the numerical PDE solution can be used as the “true” solution to compare the error between various simulation based methods we will investigate. Numerical results suggest that a particular RQMC method, called the random-start scrambled-Halton sequence, gives estimates that are sufficiently close to the PDE solution.

Numerical results show an interesting feature of the mortgage model. The endogenous mortgage rate function exhibits highly nonlinear behavior around “common” values of the interest rate. This peculiar behavior is strikingly different from the behavior of empirically modeled mortgage rates. The mortgage rates are generally higher if substantial refinancing risk is present. The “jump” represents an interest rate region which separates “low”- and “high”-refinancing risk regions. The MRB assumption (i.e., the assumption that the refinancing is driven by comparison of the mortgage rates) makes this region “accented.” This peculiarity of the endogenous mortgage rate function needs further investigation since it affects prepayment modeling and pricing (and hedging) of associated mortgage securities. A more accurate model (as opposed to using some long rate as a mortgage rate benchmark) would reduce risks to investors and, consequently, would lower mortgage rates in general.

The rest of the paper is organized as follows. We state the problem in general terms in Section 2. In Section 3 we consider a general computational algorithm in a one-factor setting and its implementation with PDE and Monte Carlo approaches. Section 4 shows numerical results for a certain specification of the prepayment rate function and CIR interest rate model. In Section 5, we study the behavior of the endogenous mortgage rate for different parameters of the prepayment

and interest rate models. We illustrate the endogenous mortgage rate function implied by Black-Karasinski and jump-diffusion interest rate models (see, for example, [2], for these interest rate models). We show that qualitative properties of the endogenous mortgage rate function stay the same as in the CIR interest rate specification. Section 7 deals with the case of heterogeneous borrowers. Section 8 discusses future research and potential approaches for the computation of the endogenous mortgage rates in a multi-factor setting. We conclude the paper in Section 9.

2 The Model

2.1 Set-up

We consider the following contract: a borrower takes a loan of P_0 dollars at the origination time t and assumes the obligation to pay scheduled coupons at rate c continuously for duration T of the contract. The loan is secured by collateral on some specified real estate property, which obliges the borrower to make the payments. The borrower has the right to settle his/her obligation and prepay the outstanding principal in a lump sum. The interest on the principal is compounded according to a contract mortgage rate m^t , which is determined at the origination time t and is fixed for the duration of the contract. Given the mortgage rate m^t and the scheduled payments c , the outstanding principal $P(u, m^t)$ (where u is time passed after the origination) can be computed.

We assume that at any time t it is possible to invest one unit in a default-free deposit account at a short-term interest rate r_t and to “roll-over” the proceeds until a later time s for a market value at that time of $e^{\int_t^s r_\theta d\theta}$. This interest rate is modeled as a deterministic function of some state variable $X_t = (X_t^1, \dots, X_t^n)$, which is assumed to be a time-homogenous Markovian process.

We consider the usual information structure described by a natural filtration of the state process $(\Omega, \mathbf{F}, \{\mathcal{F}\}_{t \geq 0}, Q)$. We interpret $\{\mathcal{F}_t\}_{t \geq 0}$ as a model of the flow of public information which is not borrower-specific. The filtration is an intrinsic feature of the market: this means that all traders have the same information available at any given time. The state process X_t represents the source of randomness of prepayments due to the economic factors affecting mortgage prices. The mortgage rate process m^t should be adapted to the filtration $\{\mathcal{F}_t\}_{t \geq 0}$.

2.2 Specification of prepayment

The prepayment time is modeled as the time of the first jump of a generalized Poisson process (the so-called Cox process). This means that the probability of prepayment is driven by some \mathcal{F}_t -intensity process γ_t . Intuitively speaking, given that a borrower did not prepay as of time t , the probability that he/she will prepay over the next “short” time interval Δt is $\gamma_t \Delta t$. Equivalently, given a “large” hypothetical pool of “homogeneous” borrowers, γ_t is the rate of prepayment (as a function of a state of economy \mathcal{F}_t) in terms of proportion (of the borrowers staying in this pool at time t) per unit of time. Given this intuition, the prepayment intensity γ_t can be regarded as the prepayment rate.

As we pointed out in the introduction, prepayment modeling is a very complicated problem. By modeling the prepayment rate as a function of a set of observable predictors, we essentially specify the probability of prepayment of a given borrower to prepay in a given time interval conditioned on some given economic state. It is a well established fact that the prepayment rates are influenced by the yield curve. If interest rates fall, then prepayment rates rise. At the same time, if interest rates are expected to go down in the future (e.g., when we have an inverted yield curve), the borrowers tend to postpone refinancing hoping to get lower mortgage rates in the future. Additionally, one might want to include the media effect (historically low mortgage rates get special attention in media and induce higher refinancing), borrower’s credit rating, education, location, loan-to-value ratio, and many other variables for better predictive ability of the prepayment model. In general, the prepayment rates depend on many factors and the yield curve might not have all the information needed for a satisfactory prepayment model. However, let us emphasize here that the most often used (“the only” in some academic models) predictor is a long (e.g., 10-year) interest rate [19] which is mostly used as a *benchmark* for the 30-year fixed rate mortgage rate.

In this paper, we are interested in a specific question: What if, instead of a benchmark, we use an endogenous mortgage rate? Therefore, in the specification of the prepayment rate we single out the dependence on the current mortgage rate. Namely, for a mortgage originated at time t the intensity (prepayment rate) at future time $s > t$ is assumed to depend on a comparison of the contract m^t and the current m^s mortgage rates. That is, we assume that the intensity is a function

of the difference² between m^t and m^s along with possible dependence on the other factors, i.e., $\gamma_s := \gamma(s - t, X_s, m^t - m^s)$, where the dependence on “ $s - t$ ” stands for time passed from the origination of the mortgage to account for possible seasonality effect.

Let us observe that it is not likely that a borrower will refinance to a mortgage with a higher mortgage rate. This suggests that the prepayment intensity does not depend on the current mortgage rate m^s , if m^s is higher than the contract mortgage rate m^t . This implies the following specification of the intensity of prepayment at time $s > t$ (where t is the mortgage origination time) which we assume for the rest of the paper:

$$\gamma_s = \begin{cases} \gamma_1(s - t, X_s, m^t - m^s), & \text{if } m^t > m^s \\ \gamma_0(s - t, X_s), & \text{if } m^t \leq m^s \end{cases} \quad (1)$$

Additionally, we assume that for fixed t and x the function $\gamma_1(t, x, \xi)$ is increasing in ξ . This means that if the current mortgage rate is lower, then the probability of refinancing is higher.

The inclusion of the refinancing predictor variable $m^t - m^s$ into the definition of the prepayment rate can be justified by the fact that the borrower’s decision to refinance is often based mainly on the comparison of the contract and current mortgage rates - there are many popular calculators on the web which give refinancing “advice” based on this comparison. Another justification is that the 30-year fixed-rate mortgage stays a popular vehicle for refinancing [10], i.e., borrowers who have a 30-year fixed-rate mortgage tend to refinance to the same kind of mortgages. We can add, however, that the popularity of adjustable-rate mortgages (ARM) is growing recently. At the same time, there is a possibility of refinancing to a 15-year mortgage. The 15-year mortgage rate process is different from the 30-year one because they depend on different terms. Though this is out of scope of this paper, we note here that both these possibilities might be included in our framework by considering the prepayment intensity as a function of not only the same kind of mortgage rate (as of the original mortgage), but 15-year and ARM mortgage rates.

²The difference is taken merely for simplicity and to avoid the introduction of additional notation, such as a “refinancing incentive” function. This might be a ratio or another function (e.g., monthly payment savings) which gives a refinancing incentive based on these mortgage rates. This function must satisfy a natural property: the function must be monotonically increasing with respect to the current mortgage rate and must be zero if the the current and contract mortgage rates are the same, i.e., we require that a lower current mortgage rate must imply a higher refinancing incentive.

One “simplification” we make for most of the paper is that we assume that borrowers are homogenous. That is, the probability of prepayment given some state of economy (the current mortgage rate in our case) is the same across all borrowers. This is known not to be the case in practice. Indeed, there is empirical evidence referred to as the *burn-out* effect: the total prepayment rate in a pool of mortgages which experienced a refinancing wave is lower because borrowers, who are more likely to refinance, did so in the time of the previous refinancing wave (the pool is “burned-out”). We address the case of heterogeneous borrowers in Section 7. The problem of the mortgage rate in the non-homogenous case is easier from a computational point of view because a solution in this case is smoother (as we will see) and, therefore, behaves better. From this viewpoint, our “simplification” (homogenous borrowers) is indeed only a notational one and can be regarded as an extreme case (computationally).

2.3 Mortgage rate equation

It can be shown (see Goncharov [7]) that the price of the mortgage M_t (originated at time t , i.e., the contract mortgage rate is m^t) can be expressed as the following expectation:

$$M_t = P(0, m^t) + \mathbf{E} \left[\int_t^{t+T} (m^t - r_u) P(u - t, m^t) e^{-\int_t^u (\gamma_\theta + r_\theta) d\theta} du \mid X_t \right]. \quad (2)$$

To exclude an arbitrage opportunity, the price of the mortgage at the origination should be equal to the initial outstanding principal, i.e., $M_t = P(0, m^t) = P_0$. Using this fact in equation (2), we obtain

$$\mathbf{E} \left[\int_t^{t+T} (m^t - r_u) P(u - t, m^t) e^{-\int_t^u (\gamma(\theta - t, X_\theta, m^t - m^\theta) + r_\theta) d\theta} du \mid X_t \right] = 0, \quad (3)$$

which can be written as

$$m^t = \frac{\mathbf{E} \left[\int_t^{t+T} r_u P(u - t, m^t) e^{-\int_t^u (\gamma(\theta - t, X_\theta, m^t - m^\theta) + r_\theta) d\theta} du \mid X_t \right]}{\mathbf{E} \left[\int_t^{t+T} P(u - t, m^t) e^{-\int_t^u (\gamma(\theta - t, X_\theta, m^t - m^\theta) + r_\theta) d\theta} du \mid X_t \right]}. \quad (4)$$

Thus, using the time-homogeneity of the Markovian state process X_t (and γ_t process), the mortgage rate process can be defined as $m^t = m(X_t)$, where the function $m(x)$ is the solution of the following func-

tional equation

$$m(x) = \frac{\mathbf{E} \left[\int_0^T r_u P(u, m(x)) e^{-\int_0^u (\gamma(\theta, X_\theta, m(x)) - m(X_\theta)) + r_\theta} d\theta \mid X_0 = x \right]}{\mathbf{E} \left[\int_0^T P(u, m(x)) e^{-\int_0^u (\gamma(\theta, X_\theta, m(x)) - m(X_\theta)) + r_\theta} d\theta \mid X_0 = x \right]} \quad (5)$$

Our objective is to find the real-valued function $m(\cdot)$ satisfying (5). For the current progress on the question of the existence and uniqueness of a solution of this equation see [8].

3 Solving the Model

We will write the right hand side of the mortgage rate equation (5) as $\mathcal{A}[m(\cdot)](x)$, where $m(\cdot)$ is the mortgage rate function, and \mathcal{A} is a functional operator. With this new notation, the mortgage rate equation (5) takes the simple form $m = \mathcal{A}[m]$, underlining the fact that the solution of the mortgage rate equation is a fixed point (function) of the operator \mathcal{A} .

The straightforward procedure of solving (5) would be to specify an initial guess as some function of x (e.g., 10-year Treasury yield) and compute the following iterations:

$$m_{i+1} = \mathcal{A}[m_i], \quad i = 0, 1, 2, \dots$$

where the function $m_{i+1}(x)$ should be computed for all x before we can move to the next iteration (since the operator $\mathcal{A}[m]$ depends on m as a function). Note that, in this case, the computation of, say, $m_1(x)$ is based on an ad-hoc initial guess. The convergence properties of this iterative scheme is currently unknown. In this section, we will discuss a modified version of this iterative scheme, which is computationally more efficient.

Our algorithm reduces the problem to a series of *scalar* equations, where iterations for each scalar equation depend on the “correct” mortgage rate obtained in the previous scalar equations. This sequential feature of the algorithm makes the computations more efficient. In this paper, we consider a one-factor state process for the instantaneous interest rate r_t . The generalization of the algorithm in a multi-factor case is discussed in Section 8. In a one-factor Markovian interest rate environment the whole yield curve is determined by the value of r_t , and thus the mortgage rate process m^t is merely a function of the interest rate r_t , i.e., $m^t = m(r_t)$.

Our algorithm uses the following lemma.

Lemma. For each interest rate value r^* , $\mathcal{A}[m(\cdot)](r^*)$ does not depend on the mortgage rate values $m(r)$ for $r > r^*$.

Proof. In [8], Goncharov proved the monotonicity of a solution to (5). Therefore, if $r > r^*$, then $m(r^*) - m(r) \leq 0$. Then the conclusion follows from our specification of the intensity process (1). \square

This lemma implies that if we know $m(r)$ for all $r < r^*$, then $m(r^*)$ can be computed by solving equation (5) as a non-linear *scalar* (not functional!) equation. Indeed, if we fix r^* , then γ_t depends on the mortgage rate function only via $m(r^*)$ because:

- 1) $m(r)$ for $r < r^*$ is fixed (known);
- 2) from the Lemma we conclude that γ_t does not depend on (unknown) $m(r)$ for $r > r^*$.

Assume that r^0 is a “low” value of the interest rate so that a mortgage originated at such a low interest rate is “immune” to refinancing due to the impossibility of getting a “cheaper” mortgage (i.e., in the case of lower future interest rates) to cover the associated transaction costs. Then m^0 can be computed as a solution of (5) with the function γ_0 (given in (1)) which does not depend on the mortgage rate process (in this case, we expect prepayment only for exogenous reasons like selling house due to divorce or relocation, etc.). That is, equation (5) is merely a non-linear *scalar* equation. This solution can be found using the fixed-point iteration: Let m_0^0 be our initial guess, and define the sequence m_i^0 , $i = 1, 2, \dots$ recursively as

$$m_{i+1}^0 = \mathcal{A}(m_i^0) = \frac{\mathbf{E} \left[\int_0^T r_s P(s, m_i^0) e^{-\int_0^s (\gamma_0(\theta, r_\theta) + r_\theta) d\theta} ds \mid r_0 \right]}{\mathbf{E} \left[\int_0^T P(s, m_i^0) e^{-\int_0^s (\gamma_0(\theta, r_\theta) + r_\theta) d\theta} ds \mid r_0 \right]}, \quad (6)$$

where m_i^0 enters the right hand side of the equation through the definition of $P(t)$ only. We conjecture that $m_i^0 \rightarrow m^0$ for $i \rightarrow \infty$. Although we have been unable to prove this conjecture, these sequences seem to converge in all the numerical experiments we have conducted.

Let $\{r^n\}_{n=0}^N$ be a grid over the interest rate values. Our purpose is to find approximations of $m^n := m(r^n)$, $n = 0, 1, 2, \dots, N$.

The mortgage rates m^n ($n = 1, \dots, N$) are defined as limits of sequences $\{m_i^n\}_{i \geq 1}$ (to be defined iteratively below). If we have found m^k for $k = 1, \dots, n - 1$, then the computation of $m^n = m(r^n)$, as was shown in the Lemma, will require only values of the mortgage rate

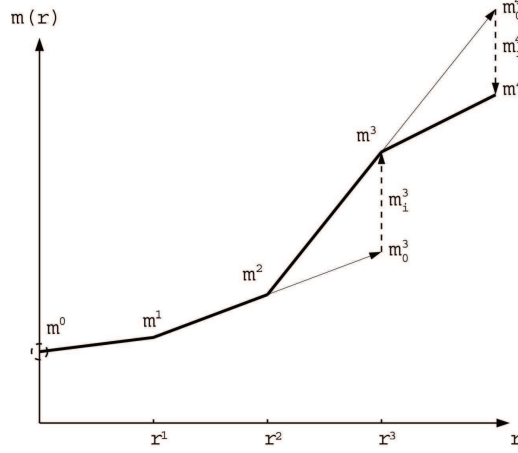


Figure 1: Consecutive application of the iterative scheme (6)

$m(r)$ for $r < r^n$. To approximate the mortgage rate function $m(r)$ when $r < r^n$, we introduce the interpolated mortgage rate function $\hat{m}_i^n(r)$ which is obtained by piecewise linear interpolation on the computed mortgage rates m^k ($k = 0, 1, \dots, n-1$) and the current iteration m_i^n (see Figure 1). Thus, we can find m^n ($n = 1, \dots, N$) inductively. When doing so, an iterative value m_i^n will depend on the “correct” (approximated) values of the mortgage rates on all the interest rate values except the interval $(r^{n-1}, r^n]$.

Now assume we know m^1, \dots, m^{n-1} . The next mortgage rate m^n is computed using the fixed-point iteration

$$m_{i+1}^n = \mathcal{A}(m_i^n) = \frac{\mathbf{E} \left[\int_0^T r_s P(s, m_i^n) e^{-\int_0^s [\gamma(\theta, r_\theta, m_i^n - \hat{m}_i^n(r_\theta)) + r_\theta] d\theta} ds \mid r_0 = r^n \right]}{\mathbf{E} \left[\int_0^T P(s, m_i^n) e^{-\int_0^s [\gamma(\theta, r_\theta, m_i^n - \hat{m}_i^n(r_\theta)) + r_\theta] d\theta} ds \mid r_0 = r^n \right]}.$$

(7)

Theoretical properties of this fixed-point iteration are not known at this point. However, our numerical results indicate convergence. The initial guess for the fixed-point iteration will be made by extrapolating from the previously computed mortgage rates m^{n-2} and m^{n-1} .

In the rest of this section we see how PDE and simulation techniques can be used to estimate the above expected values.

3.1 The PDE approach

In order to implement the iterative scheme (7), we need to estimate the expectations that appear in the equation. Let $U(r; m(\cdot))$ and $W(r; m(\cdot))$ be the numerator and denominator of the right-hand side of equation (7). For a given function $m(\cdot)$, under some technical condi-

tions (see Goncharov [7]), these functions are solutions of the following differential equations:

$$\frac{\partial U}{\partial t} + \mathcal{L}U - [r + \gamma]U + rP = 0, \quad U_T(r) = 0, \quad (t, r) \in (0, T) \times D \quad (8)$$

and

$$\frac{\partial W}{\partial t} + \mathcal{L}W - [r + \gamma]W + P = 0, \quad W_T(r) = 0, \quad (t, r) \in (0, T) \times D, \quad (9)$$

where operator \mathcal{L} is the generator of the diffusion state process r_t , i.e.,

$$\mathcal{L} := \frac{\sigma^2}{2} \frac{\partial^2}{\partial r^2} + \mu \frac{\partial}{\partial r}.$$

Thus, every iteration in (7) requires the solution of PDEs (8) and (9) with a new $m(\cdot) = \hat{m}_i^n(\cdot)$ (new $m(\cdot)$ implies new coefficients γ and P in the PDEs) evaluated at $r = r^n$. So the iterations (7) take the following form:

$$m_n^{i+1} = \frac{U(r^n; \hat{m}_i^n(\cdot))}{W(r^n; \hat{m}_i^n(\cdot))}, \quad i = 1, \dots; \quad n = 1, \dots, N.$$

For the simple one-factor interest rate model considered here, these computations are not intractable. However, if the interest rate model is complicated (for example, jump-diffusion), or has several factors, then the solution of the PDEs might get difficult, or even, intractable. Clearly, a mortgage model incorporating factors such as long and short interest rates, house prices, default, etc., will give more accurate results. Hence, in solving such multi-factor models, Monte Carlo simulation will be a good option to use.

3.2 The Monte Carlo approach

We first discretize the model in time and rewrite the recursions (6) and (7) as

$$m_{i+1}^0 = \frac{E \left[\sum_{j=1}^J r_j P(t_j, m_i^0) e^{-\sum_{k=1}^j (\gamma_0(t_k, r_k) + r_k) \Delta t} \Delta t \mid r_0 = r^0 \right]}{E \left[\sum_{j=1}^J P(t_j, m_i^0) e^{-\sum_{k=1}^j (\gamma_0(t_k, r_k) + r_k) \Delta t} \Delta t \mid r_0 = r^0 \right]}; \quad i = 0, \dots \quad (10)$$

and

$$m_{i+1}^n = \frac{E \left[\sum_{j=1}^J r_j P(t_j, m_i^n) e^{-\sum_{k=1}^j (\gamma(t_k, r_k, m_i^n - \hat{m}_i^n(r_k)) + r_k) \Delta t} \Delta t \mid r_0 = r^n \right]}{E \left[\sum_{j=1}^J P(t_j, m_i^n) e^{-\sum_{k=1}^j (\gamma(t_k, r_k, m_i^n - \hat{m}_i^n(r_k)) + r_k) \Delta t} \Delta t \mid r_0 = r^n \right]}, \quad (11)$$

where $i = 0, 1, \dots$ for $n = 0, \dots, N$, m_0^n are initial guesses, $\{t_j\}_{j=1}^J$ is the discrete time grid with number of time steps J , $\Delta t = T/J$, and r_j is the interest rate at time t_j .

The main difference between (10) and (11) is that in the second recursion the function γ depends on the previous mortgage rates

m^0, \dots, m^{n-1} as well as the rate m_i^n obtained in the “previous” step of the recursion. As we discussed earlier, m^n are computed sequentially, i.e., first m^0 , then m^1 , etc. The initial guess m_0^0 is arbitrary, m_0^1 is chosen to be $m^0 + r^1 - r^0$, and the following m_0^n ($n = 2, \dots, N$) are obtained by linearly extrapolating m^{n-1} and m^{n-2} values. The interpolated mortgage rate function $\hat{m}_i^n(r)$ is obtained by piecewise linear interpolation as discussed earlier.

We now describe in detail how to solve the first recursion numerically; the second recursion is solved similarly. We define estimators, recursively, by

$$m_{i+1,M}^0 = \frac{\sum_{l=1}^M \left[\sum_{j=1}^J r_j^{(l)} P(t_j, m_{i,M}^0) e^{-\sum_{k=0}^j (\gamma_0(t_k, r_k^{(l)}) + r_k^{(l)}) \Delta t \Delta t} \right]}{\sum_{l=1}^M \left[\sum_{j=1}^J P(t_j, m_{i,M}^0) e^{-\sum_{k=0}^j (\gamma_0(t_k, r_k^{(l)}) + r_k^{(l)}) \Delta t \Delta t} \right]}; \quad i = 0, \dots, \quad (12)$$

where $r_1^{(l)}, r_2^{(l)}, \dots, r_J^{(l)}$ ($l = 1, \dots, M$) is the l^{th} simulated interest rate path at times t_1, t_2, \dots, t_J .

The following array helps in understanding the convergence of the estimators (12):

$$\begin{array}{ccccccc} m_{1,M}^0 & m_{1,2M}^0 & \cdots & \longrightarrow & m_1^0 \\ m_{2,M}^0 & m_{2,2M}^0 & \cdots & \longrightarrow & m_2^0 \\ \vdots & \vdots & & & \vdots \\ m_{i,M}^0 & m_{i,2M}^0 & \cdots & \longrightarrow & m_i^0 \\ \vdots & \vdots & & & \vdots \end{array}$$

Each row of the array corresponds to a sequence of sample estimates obtained by increasing the sample size, say, as $M, 2M, \dots$, and hence convergence in each row corresponds to an almost sure Monte Carlo convergence. Each column of the array, on the other hand, uses the recursion (10) to compute the next value of the recursion $m_{i+1,\circ}^0$ using the previously computed value $m_{i,\circ}^0$.

In principle, we would start computing the array with the first row and take the limit as $M \rightarrow \infty$ to obtain m_1^0 . Note that the estimator $m_{1,M}^0$ is an asymptotically unbiased estimator of m_0^1 , which follows directly from its definition (12). Then we would compute $m_{2,M}^0$ using the *actual* value m_1^0 in the recursion (12), instead of the estimated value $m_{1,M}^0$. The limit of $m_{2,M}^0$ as $M \rightarrow \infty$ would similarly give us m_2^0 . This ideal process continues until a tolerance level for the fixed-point iteration is reached.

In practice, however, we compute a column of this array, say, for simplicity, the very first one:

$$m_{1,M}^0, m_{2,M}^0, \dots, m_{I,M}^0$$

Each value $m_{i,M}^0$ in this sequence is an estimate of the theoretical limit m_i^0 obtained from a Monte Carlo simulation with M sample paths. There are two sources of error in this sequence: an evaluation error in each $m_{i+1,M}^0$ due to the evaluation of the expectations, and an error that results from using the estimated value of $m_{i,M}^0$ in computing $P(t_j, m_{i,M}^0)$ in the recursion (12) for $m_{i+1,M}^0$.

A rather important part of our simulation approach is the following: the pseudorandom numbers, which we use to simulate the interest rate paths $\{r_j^{(l)}\}_{j=0,l=1}^{J,M}$, are the same for every iteration and every initial interest rate $r^n, n = 0, 1, \dots, N$. This can be viewed as a correlated sampling technique, which is a commonly used variance reduction method in Monte Carlo. Intuitively, using the same random numbers ensures that when we compute two mortgage rates at two different initial interest rates, all other factors are simulated exactly the same way. Empirically, correlated sampling results in smooth and monotonic mortgage rate curves as will be seen in the figures that follow. If we had used different random numbers, in other words, had insisted on independence across different initial interest rates, then the resulting variance of mortgage rates would have practically eliminated any hope of getting monotonic mortgage curves.

4 Implementation and Numerical Results

In this section, we use the same interest rate model and intensity function considered in Stanton [20], except for the specification of the refinancing incentive. In our case, the refinancing incentive is the difference between the contract and current mortgage rates, while in [20] it is the borrower's liability.³ The parameters of the intensity and interest rate models are taken to be close to the parameters estimated in [20].

We assume the following special case of the prepayment intensity function:

$$\gamma_t = \begin{cases} \gamma_1, & \text{if } m^0 > m^t + \delta \\ \gamma_0, & \text{if } m^0 \leq m^t + \delta \end{cases} \quad (13)$$

where γ_0 , γ_1 , and δ are constants and represent the intensity of prepayment due to exogenous reasons, the intensity of prepayment in the case when it is financially justifiable, and the transaction costs of refinancing in terms of mortgage percentage. In our base case, we assume

³This approach is called option-based in the literature.

$\gamma_0 = 0$, $\gamma_1 = 0.65$, and $\delta = 1\%$,⁴ and later discuss perturbations from these values. This choice implies that borrowers do not prepay for exogenous reasons and they check their refinancing opportunities in stochastic time intervals with an average “waiting” time of $1/\gamma_1 \approx 1.5$ years. The transaction cost of 1% implies that borrowers require about 10% of decrease on their monthly mortgage payments (if we compute these “savings” given by 1% mortgage rate decrease) to find refinancing profitable.

A general reason for choosing a step function for the prepayment intensity is an empirical fact (well-known among practitioners) that the prepayment rate curve has an S-like shape: borrowers do not prepay often when it leads to financial losses; the prepayment rates tend to grow very fast when mortgage rates fall to significantly low values; and this growth eventually stabilizes. The step-function might be considered as an extreme version of an S-like curve.

For the interest rate model, we use the one-factor Cox-Ingersoll-Ross model (see [2])

$$dr_t = \alpha(\mu - r_t)dt + \sigma\sqrt{r_t}dW_t$$

with parameters: $\alpha = 0.3$, $\mu = 0.07$, and $\sigma = 0.115$. The parameter μ has an interpretation of the long mean rate; α is a mean-reverting force; the volatility of the interest rate is assumed to be proportional to the square root of its value.

4.1 Randomized quasi-Monte Carlo methods

In this section, we give a brief overview of the randomized quasi-Monte Carlo (RQMC) methods, since a particular RQMC method will prove to be the most efficient simulation technique in the numerical results that follow. A detailed discussion of these methods can be found in [15].

We first briefly describe the quasi-Monte Carlo (QMC) method, which is the building block for RQMC. QMC methods use the so-called low-discrepancy sequences in simulating the underlying model, in contrast to the pseudorandom sequences used by Monte Carlo. Low-discrepancy sequences are deterministic sequences designed to have the best uniformity, or evenness, in their domain of definition, in contrast to the “randomness” of pseudorandom sequences. Some of the

⁴The intensity of prepayment due to exogenous reasons is around 0.035 in [20] which is much lower than 0.65 for the refinancing intensity. Our choice (zero) has little effect on the mortgage rate. We study how the mortgage rate changes with the change of parameter values in Section 5.

well-known low-discrepancy sequences are named after their discoverers: Halton, Sobol', Faure, and Niederreiter. Their work and an in-depth discussion of low-discrepancy sequences and QMC methods can be found in [14]. An advantage of QMC over Monte Carlo is its faster rate of convergence; for example, the rate of convergence for an s -dimensional numerical integration problem in QMC is $O(N^{-1}(\log N)^s)$, better than the Monte Carlo rate of $O(N^{-1/2})$.

There is, however, a drawback of the quasi-Monte Carlo method: it only provides us with a single estimate and there is no practical way of measuring the actual error of the estimate. On the other hand, in the Monte Carlo method we can generate several estimates and apply statistical techniques to assess the error. The RQMC methods are introduced to address this drawback. In a sense, they combine the best of two worlds: by appropriately “randomizing” a low-discrepancy sequence, they make “replication” possible, so that statistics can be used to measure error. And since the underlying sequences are still low-discrepancy sequences, the faster rate of convergence of QMC applies to the individual estimates. In the absence of any further assumptions on the integrand, the standard deviation of the quadrature rule based on most RQMC methods is $O(N^{-1}(\log N)^s)$, where s is the dimension. For “smooth” functions, a particular RQMC technique called “scrambled (t, m, s) -nets” can achieve a rate as small as $O(N^{-3}(\log N)^{s-1})$ (see [17]).

We now discuss a particular RQMC method that uses the so-called random-start Halton sequences. This method was introduced by Wang and Hickernell [21]. It is very simple to implement, and the numerical results reported in [21] suggest that the method ranks very favorably among other RQMC methods in error reduction. Our discussion of the method will be somewhat heuristic and narrow; we refer the reader to [21] or [15] for a formal general treatment. We start with defining the Halton sequence. The n th term of the Halton sequence in bases b_1, \dots, b_s (pairwise relatively prime integers greater than or equal to 2) is a vector of s components, denoted by

$$q^{(n)} = (\phi_{b_1}(n), \dots, \phi_{b_s}(n)). \quad (14)$$

The i th component of this vector is calculated as follows: First we write n in its base b_i expansion as

$$n = a_0 + a_1 b_i + a_2 b_i^2 + \dots + a_l b_i^l$$

where the coefficients a_j are between 0 and $b_i - 1$, and l is the greatest integer less than or equal to $\log_{b_i} n$. We then evaluate the *radical-*

inverse function in base b_i at n , i.e., compute

$$\phi_{b_i}(n) = \frac{a_0}{b_i} + \frac{a_1}{b_i^2} + \dots + \frac{a_l}{b_i^{l+1}}. \quad (15)$$

Repeating this calculation for all $i = 1, \dots, s$, we obtain the vector (14). The one dimensional projections of the Halton sequence are known as the van der Corput sequences. In other words, the one-dimensional sequence $\{\phi_{b_i}(n)\}_{n=1}^{\infty}$ is the van der Corput sequence in base b_i . The random-start Halton sequence is obtained from the Halton sequence by randomly skipping in the following sense: If $\mathbf{x} = (x_1, \dots, x_s)$ where $x_i = 0.d_0^{(i)}d_1^{(i)}\dots d_n^{(i)}$ (in base b_i), and $\phi_{b_i}^{-1}(0.d_0^{(i)}d_1^{(i)}\dots d_n^{(i)}) = m^{(i)}$ (see equation (15)) then the random-start Halton sequence corresponding to \mathbf{x} is

$$\left(\phi_{b_1}(m^{(1)} + n), \dots, \phi_{b_s}(m^{(s)} + n)\right). \quad (16)$$

Comparing (16) and (14), we see that the random-start Halton sequence is simply the Halton sequence in bases b_1, \dots, b_s skipped by the integer vector $(m^{(1)}, \dots, m^{(s)})$. Different independent random-start Halton sequences are obtained by independently choosing \mathbf{x} from the uniform distribution on $(0, 1)^s$.

We modify the random-start Halton sequence approach, in order to obtain even better accuracy. Instead of randomly skipping in the Halton sequence as we discussed above, we propose to skip in the *scrambled* Halton sequence. What is the scrambled Halton sequence? In the past, several authors observed that the uniformity properties of the Halton sequence can be improved significantly by carefully modifying the coefficients a_i in the expansion of $\phi_{b_i}(n)$ in (15). Braaten and Weller [1], Chaix and Faure [3], and Faure [5] all introduced such methods to improve the original Halton sequence. Here we will consider the approach taken by Faure in [5], where he introduces permutations σ on the digit set $\{0, \dots, b_i - 1\}$, and modifies the definition of $\phi_{b_i}(n)$ as

$$\phi_{b_i}(n) = \frac{\sigma(a_0)}{b_i} + \frac{\sigma(a_1)}{b_i^2} + \dots + \frac{\sigma(a_l)}{b_i^{l+1}}.$$

These permutations minimize the *extreme discrepancy* (a measure of uniformity) of the one-dimensional van der Corput sequences $\{\phi_{b_i}(n)\}_{n=1}^{\infty}$. When these permuted one-dimensional sequences are used to construct $q^{(n)}$ from (14), we obtain the so-called scrambled Halton sequence. For details, and the definition of the permutations, we refer the reader to [5]. Good numerical results have been obtained using scrambled Halton sequences; see [15] and [16]. In the following numerical results, we will compare the performance of the random-start Halton sequence approach with our modification (which we will call *random-start scrambled-Halton sequence* approach), by measuring how close

the mortgage rates are to those obtained by the PDE approach, in the simple one-dimensional model problem for which the PDE solution is easy to obtain.

4.2 Comparison of three simulation methods

We consider three simulation methods and two cases for each method. We compare results using the Monte Carlo (MC) method with the pseudorandom number generator `tt800` (see [13]), and two RQMC methods: random-start Halton (RS), and random-start scrambled-Halton (RSSCR) sequences. The two cases we consider are $\Delta t = 0.6$ and $\Delta t = 0.25$ years. These choices for Δt imply $J = 50$ and $J = 120$ time steps, respectively. In the rest of the paper, we will call the number of time steps simply the dimension of the problem. For both of these cases, we use $M = 100$ interest rate paths and $T = 30$ years. This means that we use 100 vectors of 50 dimensions or 100 vectors of 120 dimensions in simulating the interest rate paths. We start with an interest rate of 0% and increase it in 0.5% steps to 14.5%.

Figure 2 shows the PDE solution against one simulation using MC, RS, and RSSCR sequences for dimension equal to fifty. In the interval $[0.05, 0.06]$, we observe a sharp increase in the contract rate. This region is our real point of interest and also the region of highest error for the simulation methods.

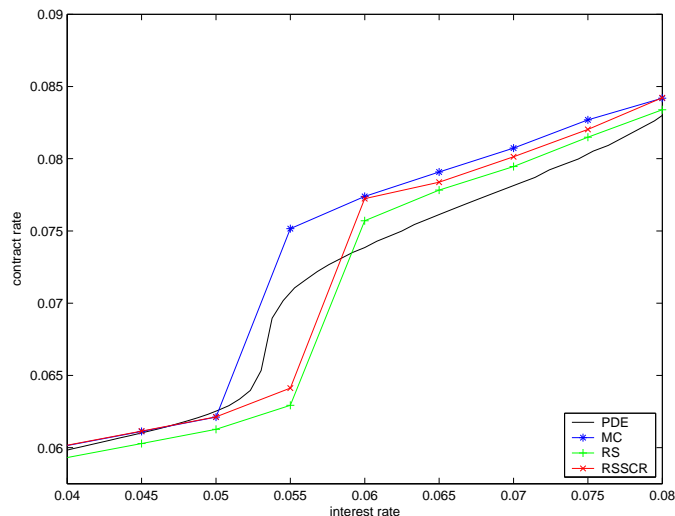


Figure 2: MC, RS, and RSSCR solutions in 50 dimensions compared to the PDE solution

Figure 3 displays the root mean square (RMS) error between the mortgage rate given by the PDE solution and the mortgage rate estimated by simulation for each initial interest rate. We generate fifty curves to compute the RMS error. The largest RMS error for MC is sixty-two basis points, while for RS it is fifty-three basis points, and for RSSCR it is forty-nine basis points. These all occur in the region of the “jump” – the interval $[0.05, 0.06]$. We notice that the starting error values are near twenty-five basis points for all methods, but RS and RSSCR begin to outperform MC as more of the curve is computed. After the jump region, RS and RSSCR have virtually the same error and both noticeably outperform MC.

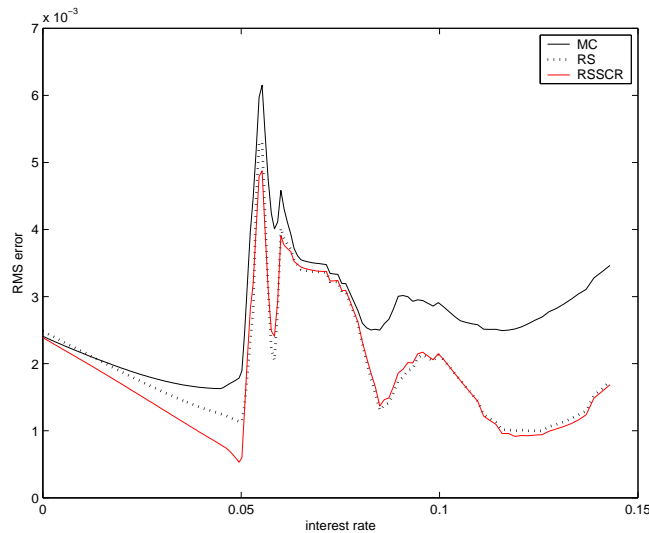


Figure 3: RMS error relative to the PDE solution

It is worth noting that in the interval $[0.00, 0.05]$ the RMS error continues to decrease for all methods. However, the errors increase dramatically for all methods after the interest rate value of 5%. This is caused primarily by the recursive nature of our scheme and the steep change in slope (the jump region). Prior to the jump, the solution is very close to linear, and hence initial guesses for the MC and RQMC methods tend to follow the slope determined by the “solved” contract rates. The RMS error for all methods is mostly in the neighborhood of or less than twenty basis points prior to the jump and in the region of the jump, the contract rate (as given by the PDE solution in Figure 2) sees an increase of over 100 basis points. This means that more often than not, the linear extrapolation used to obtain an initial guess for the contract rate being solved will be well off its mark (over fifty basis

points). This, coupled with the iterative procedure for each contract rate, will result in higher variance.

Figure 4 shows the ratio of the RMS errors relative to MC. RSSCR outperforms MC for all interest rates, while RS outperforms MC for all rates above 1.3%. However, as can be seen, the advantages of RS and RSSCR over MC diminish in the jump region. Another factor that affects the error is the dimension. The lower the dimension, the larger the partition size of the time axis and consequently a less accurate approximation for the process driving the interest rate model.

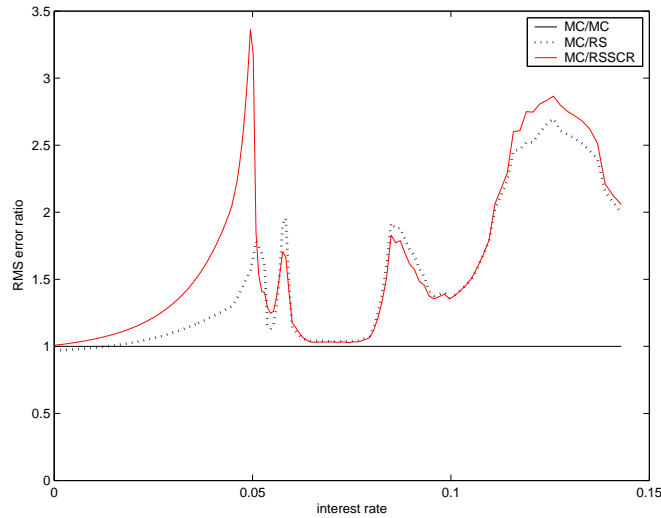


Figure 4: Ratio of values from Figure 3 compared against MC. Values greater than one imply better performance relative to MC

Figure 5 shows the PDE solution against one simulation using MC, RS, and RSSCR sequences with $\Delta t = 0.25$ years (120 dimensions). The other parameters are the same as the 50-dimensional case. From Figure 5 it is clear that the RSSCR sequence gives a better fit to the PDE solution than the other choices.

Figure 6 displays the RMS error between the mortgage rate given by the PDE solution and the mortgage rate estimated by simulation, in the 120 dimensional case. The largest RMS error for MC is forty-nine basis points, while for RS it is forty-one basis points, and for RSSCR it is thirty-six basis points. These also occur in the jump region. The error levels are lower for each method when compared to the 50-dimensional case. From Figure 6 it is evident that RSSCR dominates both methods. RS tends to be better than MC for interest rates greater than 8% but significantly worse for the initial interest rates.

Curiously enough, RS underperformed MC until the jump region, unlike in the 50-dimensional case when RS outperformed MC for virtually all interest rates.

Figure 7 and Table 1 show the ratio of the RMS error of a given method to MC RMS error. The maximum of the ratio of the MC RMS error to RSSCR RMS error is roughly 4.26, occurring at 4.95%, showing that MC has more than four times the standard deviation of RSSCR. The minimum of this ratio is approximately 1.37, showing that at worst RSSCR still has a lower RMS error than MC. On the other hand, RS does markedly worse for the lower interest rates (slightly more than two and a half times the error of MC) but outperforms MC for interest rates larger than 5%. Nevertheless, RSSCR has the lowest error of the three methods. It is worth comparing Figures 4 and 7. In Figure 4 RS and RSSCR had very small error deviations relative to each other for interest rates larger than 5%. However, when the dimension was increased from 50 to 120, RSSCR consistently outperformed RS for all interest rates. This observation is consistent with the numerical results obtained in Ökten and Srinivasan [16], where the scrambled-Halton sequence outperformed Halton sequence (without randomization) in a 100-dimensional option pricing problem.

As for computation time, there is practically no difference between the three methods. While it is true that the number of floating point operations required to generate MC numbers versus RS and RSSCR numbers is less, the difference in total processing times required is barely noticeable. The curves were generated on a 2GHz processor with 512MB of RAM. The maximum time differential in generating fifty curves using MC, RS, and RSSCR methods was about two seconds (MC was the fastest and RSSCR was the slowest), which translates to about 0.04 seconds per curve.

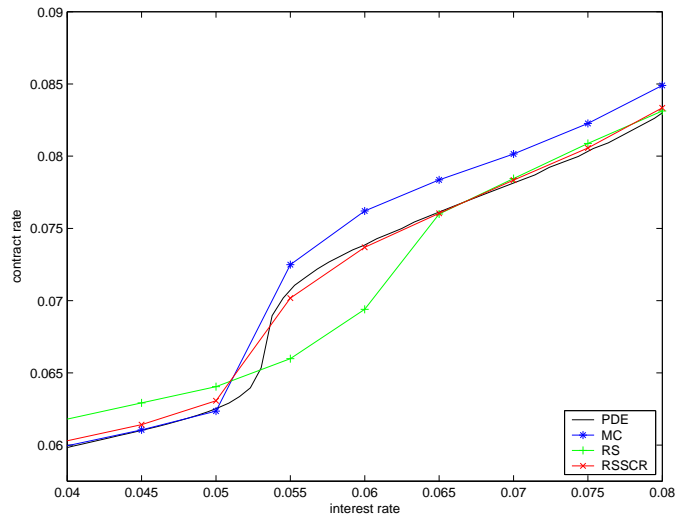


Figure 5: MC, RS, and RSSCR solutions in 120 dimensions compared to the PDE solution

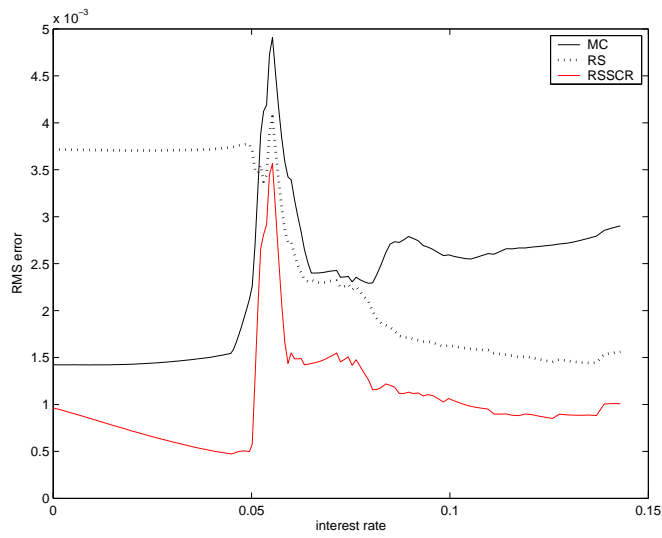


Figure 6: RMS error relative to the PDE solution

Interest Rate	MC/RS	MC/RSSCR
0.00	0.38316	1.47692
0.01	0.38320	1.68767
0.02	0.38581	1.99668
0.03	0.39288	2.42038
0.04	0.40491	2.97046
0.05	0.60029	3.99936
0.06	1.23941	2.18808
0.07	1.04562	1.59883
0.08	1.12351	1.89939
0.09	1.62968	2.47033
0.10	1.59247	2.44964
0.11	1.66046	2.78169
0.12	1.77798	2.98712
0.13	1.86130	3.06423
0.14	1.86002	2.84796

Table 1: Relative performance of RMS errors of RS and RSSCR to MC. Values greater than one imply better performance relative to MC

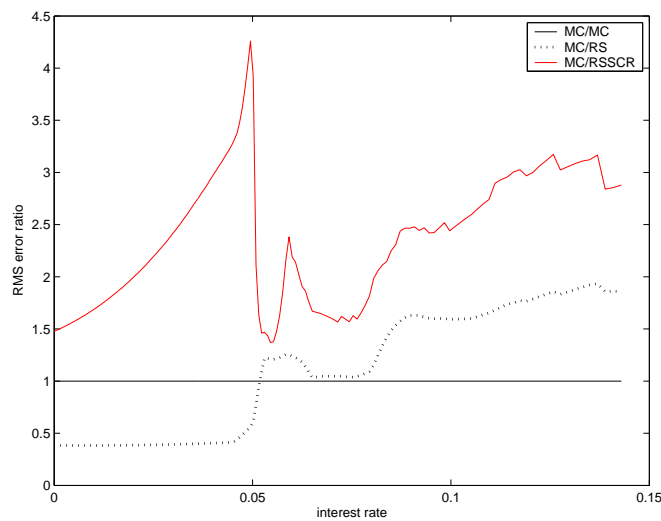


Figure 7: Ratio of values from Figure 6 compared against MC. Values greater than one imply better performance relative to MC

4.3 Dependence on number of paths

We observed that the RSSCR method was the best simulation technique among those we considered. So in this and subsequent sections, we will focus on how the RMS error is affected when one parameter is changed and all others are held constant. Figure 8 shows how the RMS error changes as the number of paths are increased. If we hold all other parameters constant then we can see that increasing the number of paths leads to a reduction in RMS error, as one would expect.

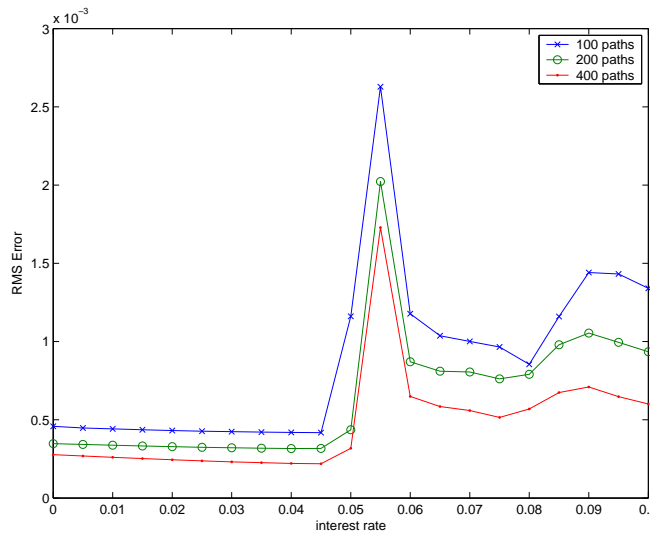


Figure 8: RMS error as number of paths changes

However, the percent reduction is not constant for a given interest rate as the number of paths are doubled. In the “usual” Monte Carlo case, we would expect to see a reduction in error by a factor of two every time the number of paths increases by a factor of four. Since we are considering an RQMC sequence, this rate of convergence is expected to be higher than the MC rate.

If we denote by $RMSE(x)$ as the RMS error for $100x$ number of paths, then the ratio of $RMSE(y)/RMSE(x)$ tells us the percent improvement (or degradation) of going from $100x$ paths to $100y$ paths. Figure 9 compares $RMSE(2)/RMSE(1)$ and $RMSE(4)/RMSE(2)$. We see that when the interest rate is 0%, increasing the number of paths from 100 to 200 leads to a 25% ($1 - .75$) decrease in the RMS error, but increasing from 200 to 400 paths leads to a 20% ($1 - 0.8$) decrease in RMS error. We see this more noticeably at an interest rate of 5%. When the number of paths is increased from 100 to 200 the RMS error

decreases by almost 65%, but increasing from 200 to 400 paths, we have a more modest decrease of roughly 25%.

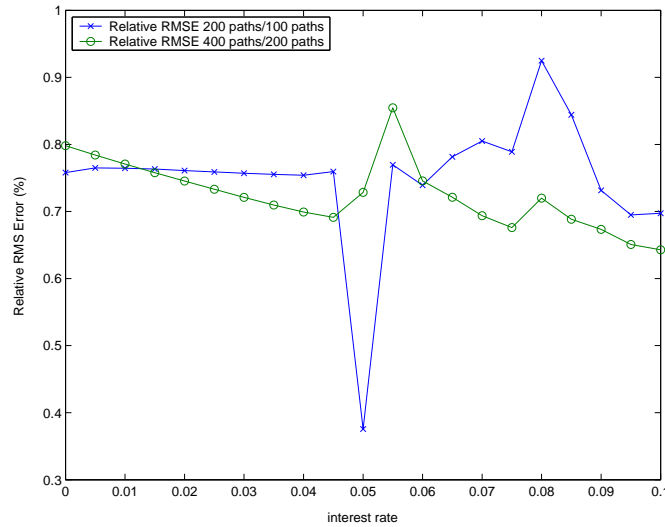


Figure 9: Relative RMS error

4.4 Dependence on interest rate grid

Here we discuss how the choice for the interest rate grid, r^0, r^1, \dots, r^N , affects the mortgage rate curve. If the interest rate grid is too coarse then changes in the slope of the mortgage rate curve will not be adequately captured, resulting in larger estimation error. If the interest rate grid is too fine, then there is a considerable waste of calculation in the region in which the mortgage rate curve is linear. What we should expect, though, is that the finer the interest rate grid, the better the estimation of the mortgage rate. We consider three uniform interest rate grids with $\Delta r = 1\%$, $\Delta r = 0.5\%$, and $\Delta r = 0.25\%$ with r ranging from $r^0 = 0\%$ to $r^N = 10\%$. Figure 10 shows the RMS error for different interest rate grids. The mortgage rate curve is constructed using the different interest grids, but we compute RMS errors only for the interest rates that coincide with the $\Delta r = 1\%$ grid, since this is the coarsest interest rate grid. We have noticeable improvement in the RMS error as we go from the 1% grid to 0.5% grid, however, there is not much improvement as the grid size is further reduced to 0.25%.

5 Case Study

5.1 Dependence on CIR parameters

We now consider the dependence of the endogenous mortgage rate on the parameters of CIR interest rate model. First, we observe the behavior of the mortgage rate curve as we vary the mean reversion coefficient, α . Figures 11 and 12 show the behavior of the mortgage rate when α increases from $\alpha = 0.10$ to $\alpha = 0.55$ in increments of 0.05.

We can see that as α increases the location of the jump moves from left to right. The mortgage rates in the “no prepayment” region (left of the jump) increase as α increases, while after the jump we have the mortgage rate decreasing. Additionally, the interest rate domain of the no prepayment region gets larger as α increases. This is what we would expect, since the higher the α , the faster the reversion to the mean. For initial interest rates well below the mean, the interest rates and thus mortgage rates are expected to rise. The reverse behavior is true for interest rates greater than the mean, as higher α means that interest rates are expected to fall.

Next, we look at the behavior of the mortgage rate curve as we vary the long-term mean, μ . Figures 13 and 14 are the mortgage rate curves when μ ranges from 0.03 to 0.12 in increments of 0.01. We see that as μ increases, the height of the “jump” steadily decreases and also smooths out. For large μ we see that the prepayment and the no prepayment region come together. This is what one would expect, since if the long-term mean increases, the incentive to prepay mortgages occurs only at higher contract rates.

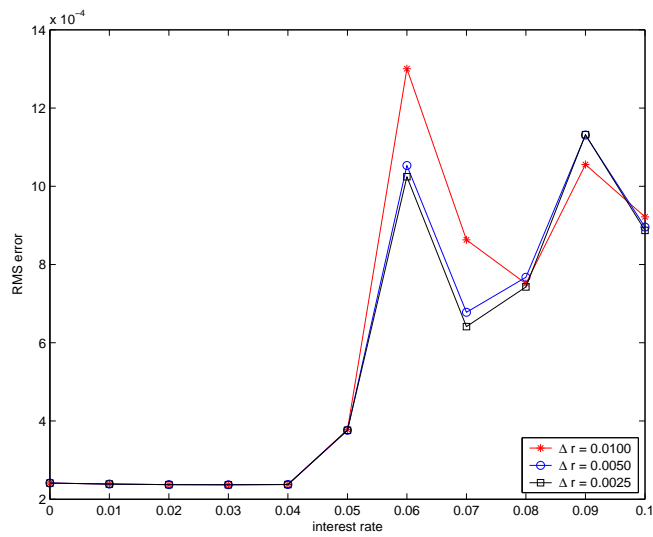


Figure 10: RMS error for different interest rate grids

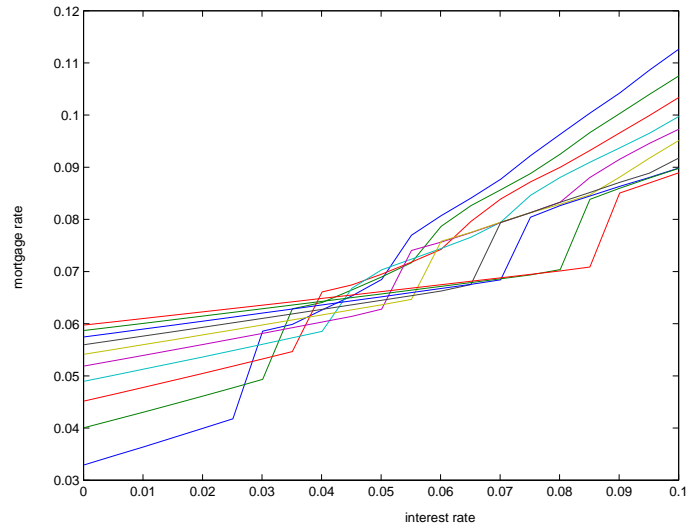


Figure 11: Sample curves for varying α

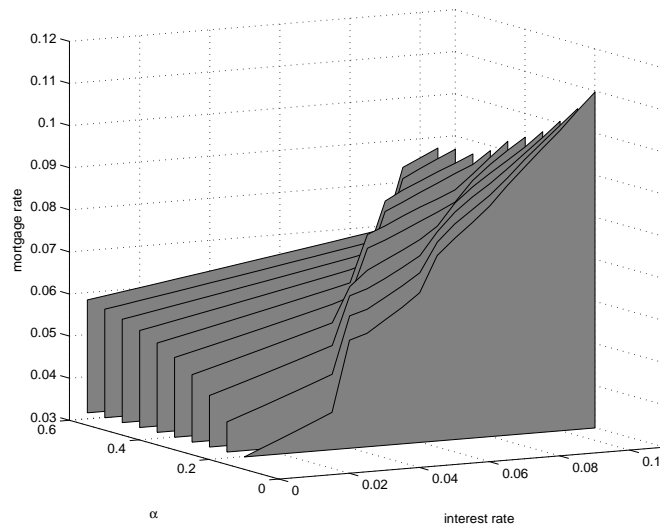


Figure 12: 3D representation of the effects of changing α

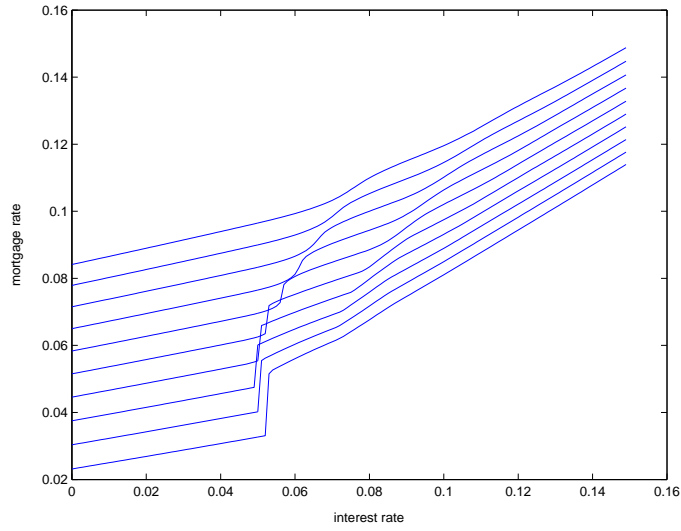


Figure 13: Sample curves for varying μ

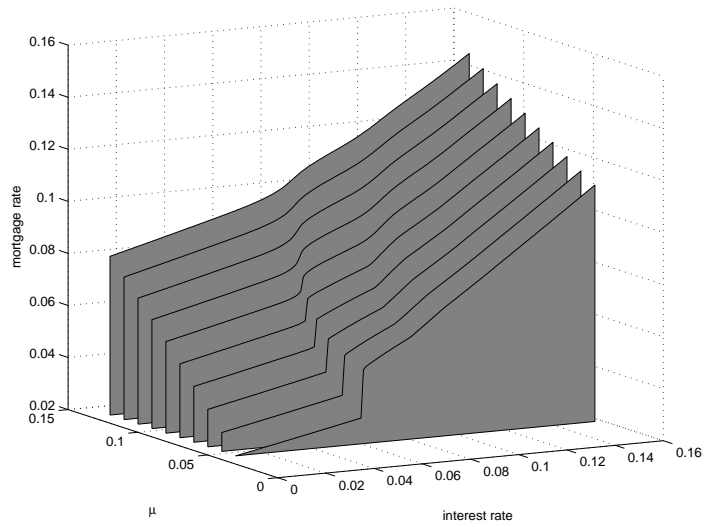


Figure 14: 3D representation of the effects of changing μ

5.2 Dependence on intensity parameters

Here we consider the effect of the intensity (intensity of prepayment for exogenous reasons, intensity of refinancing, and transaction costs) on the mortgage rate curve.

Figures 15 and 16 show the mortgage rate curve for different choices of γ_0 . The increase in γ_0 value smooths the mortgage rate function. The γ_0 intensity is responsible for prepayment due to exogenous reasons in situation when the prepayment is “desirable” for investors, i.e., this prepayment occurs when the interest rates are higher.⁵ Therefore, the mortgage rate is generally lower. Note, however, increase of the mortgage rate for the values of the interest rate just on the left of the “jump.” This peculiar behavior is the product of functional dependence in the endogenous mortgage rate equation. When the mortgage rate function smooths out (and the mortgage rates decrease more noticeably for the low interest rate values), the refinance/no refinance regions diffuse and the “jump” becomes less accented than in the case of small values of γ_0 . This “diffusion” is responsible for increasing refinancing risk for the interest rate values around 5% and, consequently, for the higher mortgage rates.

Figures 17 and 18 show the mortgage rate curve for different choices of γ_1 (rational prepayment). The larger the value of γ_1 , the higher the likelihood of prepayment if the interest rate falls into the “refinance” region. Therefore the refinancing risk (which has an adverse effect on the mortgage investors) is higher. This is reflected in a steeper change of the mortgage rate when the interest rate moves from no prepayment region to prepayment region (the mortgage rate is higher there to compensate for this refinancing risk). The mortgage rate function on the left of the “jump” stays untouched since it is a “no refinance” region, i.e., mortgages originated in that low interest rate environment will not be refinanced because of the low contract mortgage rate and high transaction/refinancing costs.

Figures 19 and 20 show the mortgage rate for transaction costs varying from 0% to 2.25% in increments of 0.25%. The changes to the mortgage rate curve as transaction costs increase are evident. We see an ever increasing pronounced difference between the no prepayment region and prepayment region. The higher the transaction costs the more prohibitive it is to prepay. When transaction costs are 0%,

⁵At the same time, the increase of γ_0 might be viewed as increase of the exogenous prepayment rate for all interest rate values with simultaneous decrease of the pure refinancing intensity $\gamma_1 - \gamma_0$.

one would find refinancing profitable *every* time mortgage rates fell below the contract rate (as there is no cost in doing so). However, as transaction costs become more significant, prepaying the mortgage becomes more prohibitive even for interest rates higher than the long term mean.

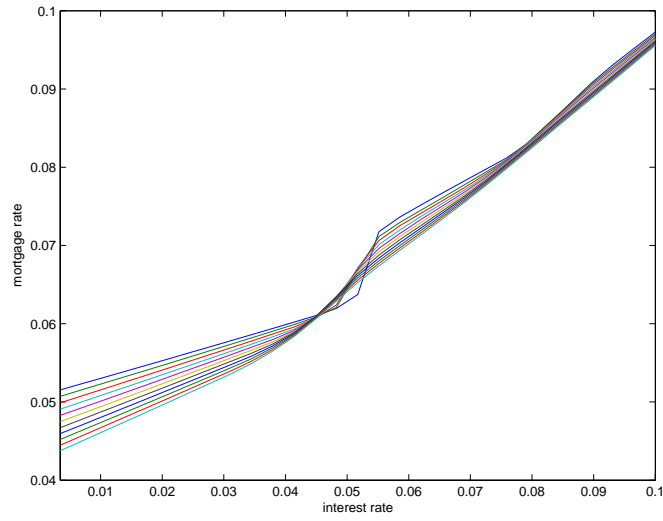


Figure 15: Sample curves for varying γ_0

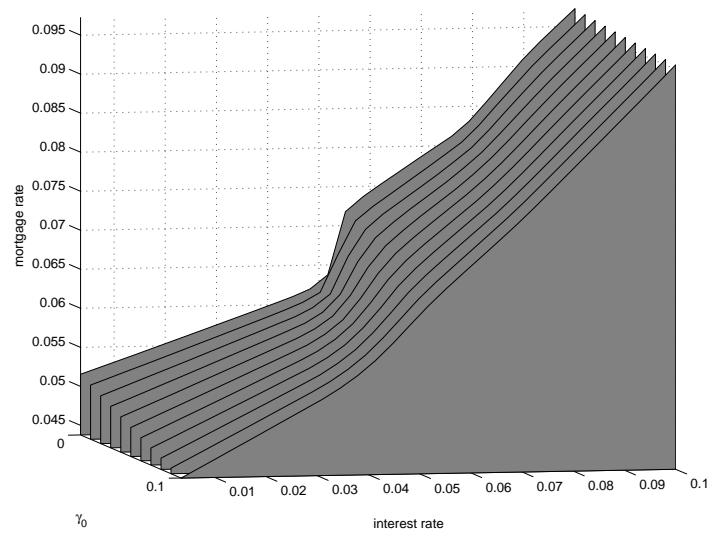


Figure 16: 3D representation of the effects of changing γ_0

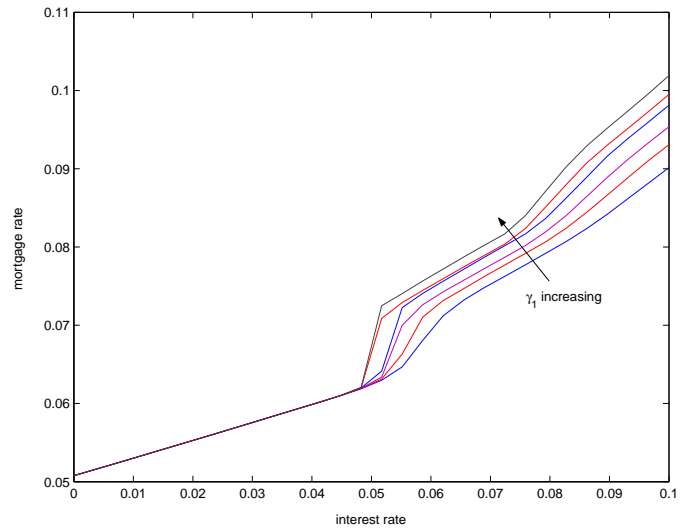


Figure 17: Sample curves for varying γ_1

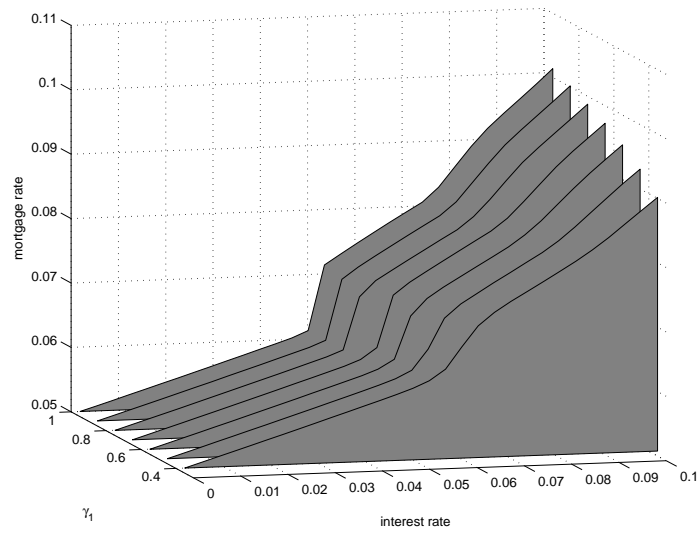


Figure 18: 3D representation of the effects of changing γ_1

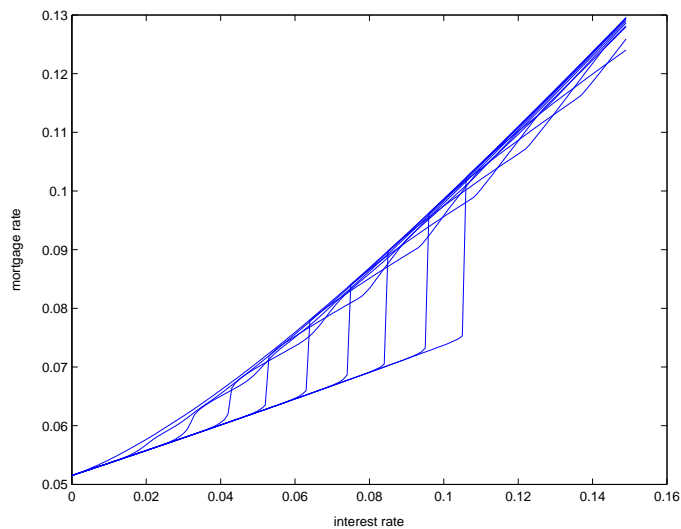


Figure 19: Sample curves for varying transaction costs

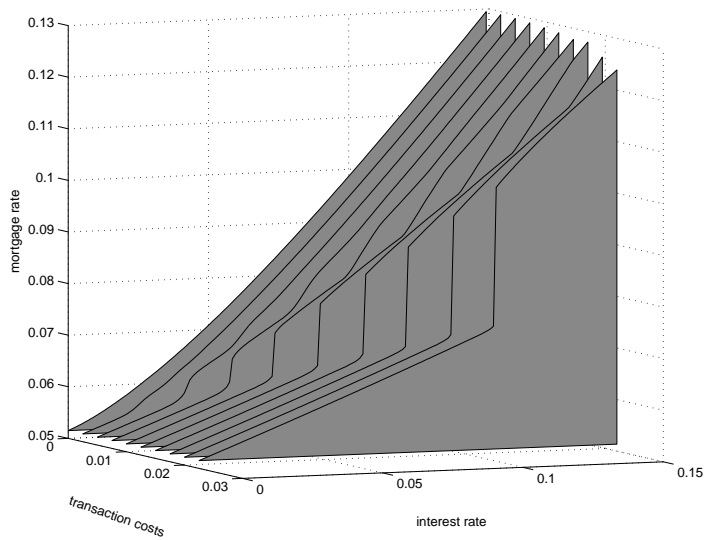


Figure 20: 3D representation of the effects of changing transaction costs

6 Alternative Interest Rate Models

In this section we consider two alternative interest rate models; the CIR model with a jump process, and the Black-Karasinski model (see [2]). We observe that the qualitative behavior of the mortgage rate function is similar in these models.

The CIR model with a jump-diffusion process is given as follows:

$$dr_t = \alpha(\mu - r_t)dt + \sigma\sqrt{r_t}dW_t + dq. \quad (17)$$

Here the dq term is the jump-diffusion process governed by a Poisson process. One specifies jumps of a certain height (or the height may come from another distribution) which have interarrival times distributed as exponential random variables with mean λ .

The Black-Karasinski model is given as follows: we take an Ornstein-Uhlenbeck process

$$d \ln r_t = (\theta - a \ln r_t)dt + \sigma' dW_t, \quad (18)$$

where θ , a , σ' are constants.

Figure 21 shows one sample mortgage rate curve using CIR, CIR with jump, and Black-Karasinski interest rate models.⁶ We observe that the curve is very similar in each case.

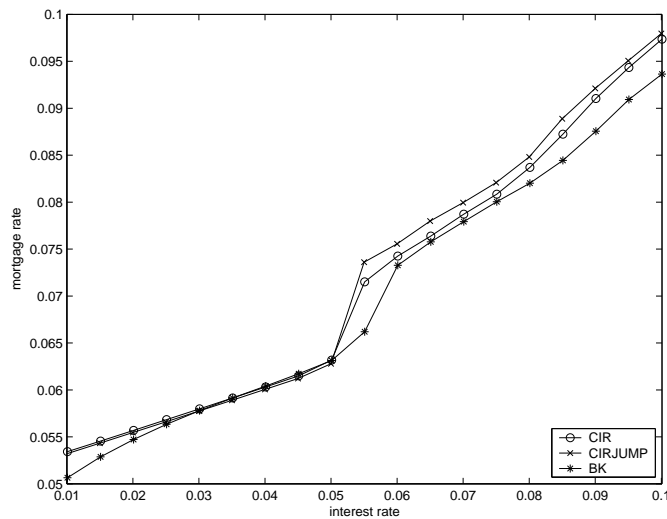


Figure 21: Sample curves from a number of interest rate processes

⁶We chose the parameters of CIR with diffusion and Black-Karasinski models in such a way that they produce similar 10-year Treasury yields.

7 Heterogeneous Borrowers

In this section, we discuss the extension of our framework to the case of pools with heterogeneous borrowers. Assume that the set of all possible borrowers' intensity functions is given by the parameterized family $\{\gamma_t^\omega\}_{\omega \in \Omega}$. Let us assume that Φ_ω is the distribution of borrowers in a mortgage pool. Following the derivation given in Section 2.3, we conclude that the mortgage rate is given by the equation

$$m(x) = \frac{\mathbf{E} \left[\int_0^T r_u P(u, m(x)) \left(\int_{\Omega} e^{-\int_0^u \gamma^\omega(m(x) - m(X_\theta)) d\omega} d\Phi_\omega \right) e^{-\int_0^u r_\theta d\theta} du \mid X_t = x \right]}{\mathbf{E} \left[\int_0^T P(u, m(x)) \left(\int_{\Omega} e^{-\int_0^u \gamma^\omega(m(x) - m(X_\theta)) d\omega} d\Phi_\omega \right) e^{-\int_0^u r_\theta d\theta} du \mid X_t = x \right]}.$$

(19)

This equation can be solved following the same algorithm discussed in Section 3, once the integral with respect to the distribution Φ_ω is approximated using a quadrature rule.

In Figure 22, we illustrate a solution to this equation assuming that all borrowers can be divided into three (with equal proportions) groups, with three different transaction costs; 0.7%, 1% and 1.3%. The other parameters are taken the same as in Section 4.

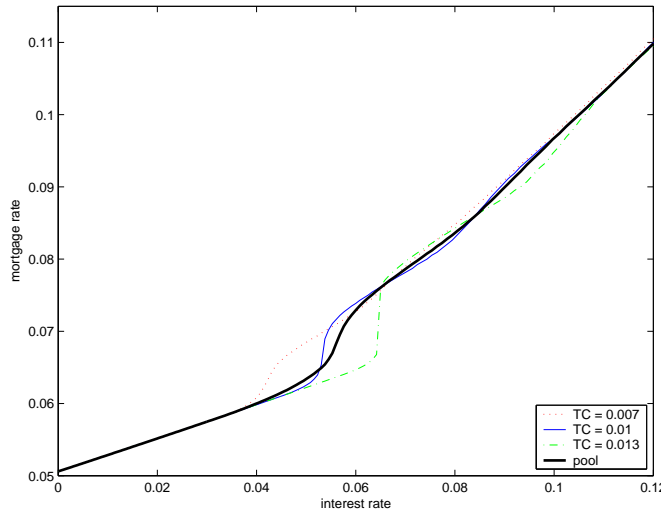


Figure 22: The mortgage rate in the case of heterogeneous borrowers

As we can see, in the presence of heterogeneous borrowers the “jump” region is smoothed out. Computationally, it is reflected in the lower spike in root mean square error than in the case of homogenous borrowers (compare Figure 23 with Figure 3).

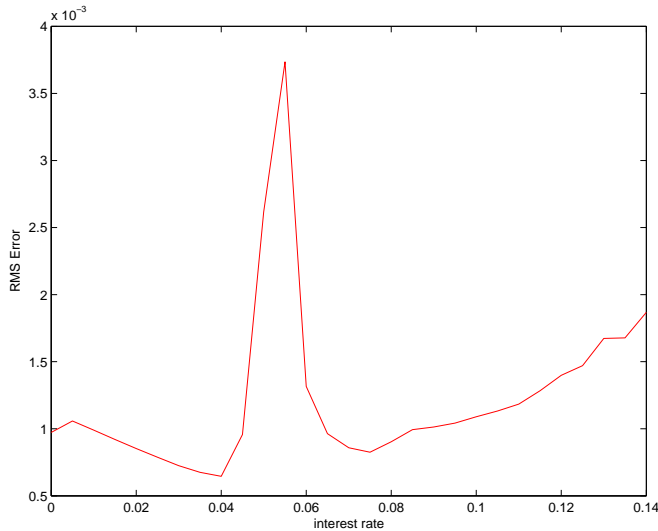


Figure 23: RMS error of RSSCR against the PDE solution for the heterogeneous borrowers case

8 Future Research; Multi-factor Extensions

A critical feature of our MC algorithm is the ordering of the interest rate variable and using the monotonicity of the mortgage rate function. In a multi-factor interest rate model (or, a multi-factor state process in general), however, this ordering is no longer applicable since the interest rate is now a multi-dimensional vector. However, ordering can be applied to the mortgage rate space since it is one-dimensional. Namely, instead of discretizing the interest rate factors, we discretize the (one-dimensional!) values of the mortgage rates m^n , $n = 0, \dots, N$. Then for every (fixed) m^n , the equation (5) is solved for the interest rate factors (in an increasing order of n). Solutions, which are now the level curves $R^n = \{(x, y) \mid m^n(x, y) = \mathcal{A}[m^n](x, y)\}$, are found using regression based Monte Carlo techniques, similar to those used in [12]. This procedure allows us to use the monotonicity of the mortgage rate function as was done in this paper: the level curve R^n defines the boundary of the domain of dependence for the mortgage rate (the mortgage rates are higher “outside” R^n and does not influence the prepayment decision). Thus, the solution R^n can be found based only on

“previous” level curves R^k , $k < n$ (more precisely, on the interpolation of the surface based on these level curves) and, therefore, on “correct” values of the previously found mortgage rates on these curves. This extension is proposed in Goncharov [9] and encouraging preliminary results are obtained.

The above approach promises to be an efficient generalization of our present MC algorithm. However, there is another MC approach which we briefly mentioned in the second paragraph of Section 3. This approach easily generalizes to multi-factor problems. It solves the fixed-point problem (5) using simultaneous iterations. In other words, the algorithm does not use the monotonicity of the mortgage rate function, and its implementation is straightforward: given m_i^n , $n = 0, \dots, N$, compute the iteration m_{i+1}^n for all $n = 0, \dots, N$ by using solely the values m_i^n from the previous iteration i using (7). In this case the iterations are done over the whole mortgage rate surface “simultaneously.” Note that the first iteration, for example, is based on an ad-hoc initial guess, while our (monotonicity based) approach allowed us to find the mortgage rate for a given r^n based on the “correct” values of the mortgage rates $m(r^j)$, $j < n$. To make the computations more efficient in the case of simultaneous iterations, we can adapt a multigrid approach where the first iteration is made on a coarse mesh, and consecutive iterations are computed on refining meshes that concentrate more mesh points in the “jump” region of the mortgage rate curve. The implementation and comparison of these two MC approaches is currently under investigation by the authors.

9 Conclusion

We considered the problem of determining the mortgage rate curve as implied by the underlying interest rate process when prepayment is influenced by the mortgage rate process itself. A model introduced by Goncharov [6] writes the mortgage rate function as the fixed-point of a functional equation. Here we presented a Monte Carlo algorithm that solves this fixed-point problem. The algorithm was implemented using a randomized-quasi Monte Carlo sequence. In a one-factor setting where the PDE solution can be computed easily, we compared our implementation with the PDE solution. The results were impressive: the Monte Carlo implementation produces fast and “financially” indistinguishable results from the PDE solution. We also conducted case studies to investigate the dependence of the mortgage rate on the parameters of the model. An interesting feature of the endogenous rate in

the form of a “jump” was observed for different underlying interest rate models. This suggests that currently used benchmarks (e.g., 10-year Treasury yield) for mortgage rate process might not be adequate.

References

- [1] Braaten, E., Weller, G., *An Improved Low-Discrepancy Sequence for Multidimensional Quasi-Monte Carlo Integration*, J. Comp. Physics, 33, pp. 249–258, 1979.
- [2] Brigo, D., Mercurio, F., *Interest Rate Models - Theory and Practice*, Springer-Verlag, 2001.
- [3] Chaix, H., Faure, H., *Discrepance et diaphonie en dimension un*, Acta Arithmetica, LXIII, pp. 103–141, 1993.
- [4] Dunn, K. and J. McConnell, *Valuation of gnm mortgage-backed securities*, Journal of Finance, 36, pp. 599–617, 1981.
- [5] Faure, H., *Good Permutations for Extreme Discrepancy*, Journal of Number Theory, 42, pp. 47–56, 1992.
- [6] Goncharov, Y., *Mathematical Theory of Mortgage Modeling*, Ph.D. dissertation, University of Illinois at Chicago, 2003.
- [7] Goncharov, Y., *An Intensity-Based Approach to the Valuation of Mortgage Contracts and Computation of the Endogenous Mortgage Rate*, International Journal of Theoretical and Applied Finance, 9(6), pp. 889–914, 2006.
- [8] Goncharov, Y., *On Some Properties of the Endogenous Mortgage Rate Function*, working paper, 2006.
- [9] Goncharov, Y., *On Computation of Endogenous Mortgage Rate*, technical report, 2006.
- [10] Hayre, L. and A. Rajan, *Anatomy of Prepayment*, Advanced Fixed Income Valuation Tools (N. Jegadeesh & B. Tuckman, Eds.), John Wiley, New York, 2000.
- [11] Kau, J.B., Keenan, D.C., W.J. Muller III, and J.F. Epperson., *Generalized Valuation Model for Fixed-rate Residential Mortgages*, Journal of Money, Credit and Banking, 24(3), pp. 279–299, 1992.
- [12] Longstaff F., Schwartz E. *Valuing American options by simulation: a simple least squares approach* Review of Financial Studies, 14, pp. 113–147, 2001.
- [13] Matsumoto, M. and Kurita, Y., *Twisted GFSR generators II*, ACM Transactions on Modeling and Computer Simulation, 4, pp. 254–266, 1994.
- [14] Niederreiter, H., *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.

- [15] Ökten, G. and Eastman, W., *Randomized quasi-Monte Carlo methods in pricing securities*, Journal of Economic Dynamics & Control, 28, pp. 2399–2426, 2004.
- [16] Ökten, G. and Srinivasan, A., *Parallel Quasi-Monte Carlo Applications on a Heterogeneous Cluster*, Monte Carlo and Quasi-Monte Carlo Methods 2000, K. T. Fang, F. J. Hickernell and H. Niederreiter, Eds., pp. 406-421, Springer-Verlag, Berlin, 2002.
- [17] Owen, A.B., Scrambled net variance for integrals of smooth functions. Annals of Statistics, 25, pp. 1541–1562, 1997.
- [18] Pliska, S., *Mortgage Valuation and Optimal Refinancing*, Stochastic Finance, Proceedings of a 2004 conference in Lisbon, Portugal, edited by A. Shiryaev et al., Springer, 2006.
- [19] Schwartz, E.S. and W.N. Torous, *Prepayment and the Valuation of Mortgage-Backed Securities*, Journal of Finance, 44(2), pp. 375–392, 1989.
- [20] Stanton, R., *Rational prepayment and the valuation of mortgage-backed securities*, Review of Financial Studies, 8, pp. 677–708, 1995.
- [21] Wang, X. and Hickernell, F. J., *Randomized Halton sequences*, Mathematical and Computer Modelling, 32, pp. 887–899, 2000.