Reviewing Monte Carlo Markov Chains and demonstrating their application in Bond Pricing

Alexander K. Bechler

Abstract:

Following the systemic collapse of the banking system during the 2008 Great Financial Crisis, the broad-based use of statistics in quantitative finance and risk management has gained increasing popularity. The causes leading up to the general failure of several leading lending institutions were in part caused by the inability of institutions to understand the risk profile and exposure generated by the complexity of the products that were being offered. An alarming amount of leverage began building in the mid-2000s and continued through to the end of the crisis, with borrowing jumping to 12.8% of GDP in 2007 from 6.9% in 1997 [1]. As a result of the quagmire in the financial sector, a series of significant reforms were proposed in the form of a Comprehensive Capital and Analysis Review ('CCAR') through the Dodd Frank Act Stress Tests [2]. This act introduced a structured modeling framework that lending and trading institutions (or 'Globally Systemic Important Banks') would need to implement for regulators, shareholders, and the public to better understand a bank's risk profile. Over time, regulators have begun to increasingly accept more sophisticated analytical and modeling techniques – going from only accepting simple regression models to endorsing the use of machine learning and artificial intelligence [2]. Monte Carlo Markov Chain (MCMC) is one of the statistical techniques that have begun to garner more attention as being a way to model and understand risk in a bank's balance sheet. This review will introduce MCMC methods, demonstrate their use in financial applications, and walk through an example of their application.

Literature Review, History, and some theory:

Monte Carlo Markov Chain (MCMC) is a statistical sampling method used to draw a sample from a probability distribution of an underlying random variable. MCMC methods are unique in that they merge both Monte Carlo methods and Markov Chain methods to produce their results [3]. MCMC depends on the statistical properties of both Monte Carlo methods and Markov Chain methods. This report will review literature and theory on each statistical method independently before exploring their combined application.

Monte Carlo Simulation:

Monte Carlo methods are extremely popular in the modern era of mathematical finance and financial engineering. Simply put, Monte Carlo methods involve taking a real-world problem, finding a probability analog, defining the parameters of that analog, and then performing the simulation many times to generate a range of outcomes [3]. The range of outcomes of the simulation allow a researcher to understand the results or eventual outcomes of their real-world problem. Monte Carlo simulation was initially used in World War II, when American scientists were working on the development of the nuclear bomb in the Manhattan project. John von Neumann and Stanislaw Ulam generated a method to measure neutron travel as it passed through a radiation shield. They chose to name the simulation after the Monte Carlo Casino in Monaco. The effectiveness of Monte Carlo simulation became immediately clear. It allowed researchers, scientists, and experimenters to explore highly complex systems and to reproduce experiments and results with relative ease [4]. It also created an effective 'safety barrier' for researchers by allowing them to understand the outcomes of potentially hazardous experiments without having to conduct those experiments. Before Monte Carlo simulation, these complex problems and systems were difficult (or even impossible) to model due to their analytical complexity and safety concerns [5].

While the advantages are clear, Monte Carlo simulations also have some notable disadvantages. The predictions generated by Monte Carlo simulations are estimates and probabilities and are not exact or precise estimates [4]. As more simulations are conducted, the estimates will converge to their true value due to the law of large numbers. However, like error in sampling from a population, Monte Carlo simulations will also consistently have some degree of error in their results. Since Monte Carlo simulations require an understanding of the system that the user is trying to model, they typically require large amounts of data to generate the distributions of the system [5]. As with most research, gathering accurate and complete data can be extremely difficult, expensive, and highly limiting based on the costs involved in its procurement. Users would need to balance the cost and benefit when designing a simulation.

Markov Chains:

A Markov chain is a type of stochastic model that has also become widely used in credit risk modeling, portfolio management, interest rate risk management, and asset pricing models. Markov chains were first introduced by a Russian Mathematician named Andrei A. Markov in 1907. In his landmark work, titled "The Law of Large Numbers and Their Distribution, Dependent on Each other", Markov analyzed sequences of events (or process) in which the probability that the type of state the process moved to depends on the current state of the system [7]. Each chain had *n* number of steps that could be decided by the user. What the system transitioned to next was independent of what it was in a prior state, commonly known as the 'memoryless' property of Markov chains [6]. Given the memoryless property of Markov chains, they are generally described as a directed random walk over time, where the probabilities of the next value do not depend in any form on prior outcomes [6,7]. The current state of a Markov process is independent of the prior states of that Markov process and only depends on the current state of the process.

A simple example of this memoryless property is the arrival times of a train while you wait at a train station. Let's say a passenger arrives and waits for a train and that a train arrives every forty-five minutes according to a Poisson process. Ten minutes elapse and the passenger is still waiting for the train. The memoryless property of the Markov chain stochastic process indicates that the probability the passenger will wait for forty-five more minutes is the same as if the passenger had just started waiting for a train. This property can be mathematically expressed as the following:

$$\mathbf{Pr}(\textit{waiting time} > \textit{time spent waiting} + \textit{additional time spent waiting} \mid \textit{waiting time} > \textit{time spent waiting})$$
$$= \mathbf{Pr}\,(\textit{waiting time} > \textit{additional waiting time})$$

Transitioning to the next state of a Markov chain is determined through a series of probabilities presented in a transition (or 'design') matrix [7]. This transition matrix contains all probabilities that the current state of the matrix transitions into any of the next steps in the matrix. Suppose a weatherman wanted to use a Markov chain to forecast the weather over the next day. They would need to generate the probabilities of the system transitioning into different types of weather conditions for the next day, given the conditions of today. For example, if today is a cloudy day, here are the transitions probabilities a weatherman needs to determine:

Pr (*cloudy tomorrow* | *cloudy today*)
Pr (*sunny tomorrow* | *cloudy today*)
Pr (*rainy tomorrow* | *cloudy today*)
Pr (*snowy tomorrow* | *cloudy today*)

Over time, the output of a Markov chain is expected to converge to its prior distribution. In this example, if a fifth of the days that follow a cloudy day are sunny, then the Markov chain is expected to produce an output that is aligned with that observation if the transition matrix is properly defined. This is due to the property of stationarity inherent to a Markov chain.

The simplicity, versatility, and interpretability of Markov chains are major advantages in deploying their use. It is easy for a layperson to understand what is happening in our example: Tomorrow's weather is being modeled as a function of the weather today. However, this is not a complete list of transition probabilities for our example. What if today is sunny, rainy, or snowy? The weatherman would need to populate the design matrix with a complete list of those probabilities, shown below:

Pr (*cloudy tomorrow* | *cloudy today*)
Pr (*sunny tomorrow* | *cloudy today*)
Pr (*rainy tomorrow* | *cloudy today*)
Pr (*snowy tomorrow* | *cloudy today*)
Pr (*cloudy tomorrow* | *sunny today*)
Pr (*sunny tomorrow* | *sunny today*)
Pr (*rainy tomorrow* | *sunny today*)
Pr (*snowy tomorrow* | *sunny today)*
Pr (*cloudy tomorrow* | *rainy today*)
Pr (*sunny tomorrow* | *rainy today*)
Pr (*rainy tomorrow* | *rainy* today)
Pr (*snowy tomorrow* | *rainy today*)
Pr (*cloudy tomorrow* | *snowy today*)
Pr (*sunny tomorrow* | *snowy today*)
Pr (*rainy tomorrow* | *snowy today*)
Pr (*snowy tomorrow* | *snowy today*)

Now the weatherman has a more complete design matrix of transition probabilities, which also highlights one of the challenges of Markov chains – determining the transition probabilities of moving from one state to the next. The example presented here was trivial. It only had four possible states that were generalized and easy to determine. What if a temperature component was added and the weatherman wanted to see probabilities of transitioning to hot, mild, or cold days for each of the precipitation types? That would generate additional transition probabilities and therefore increase the size of the transition matrix. Modern computing has helped to mitigate this problem, but for systems that are increasingly large and complex it is not challenging to imagine a scenario that might produce an extremely large transition matrix [7]. The increasingly large and complex design matrix makes it harder to capture all possible transition states in a stochastic system.

Additionally, Markov chains assume stationarity in their transition probabilities [6]. In our weatherman example, a Markov chain assumes that the transition probabilities between weather types remain the same over time. External impacts and forces, however, could change these transition probabilities over time. If the earth begins a period of warming, for our example, the probability of sunny days could increase with time. Using a Markov chain to model a non-stationary process would require frequent updating of the transition probabilities or exploration of different models that account for non-stationarity.

Monte Carlo Markov Chains

Monte Carlo Markov Chains (MCMC) have been around for many years and are nearly as old as Monte Carlo simulations that were designed in World War II. The first time an MCMC was used was in 1952 at the Los Alamos laboratory in New Mexico by Nicolas Metropolis. Among other things, Metropolis and his research partners established that a step-by-step random walk of a particle transitioning through multiple spaces did end up being stationary [8]. Later, researcher Hastings further expanded on the method that Nicolas Metropolis had defined. Hastings was able to overcome the limitation of dimensionality identified by Metropolis in his paper and defined the first generalization of the algorithm defined by Metropolis [8]. This groundbreaking research by Hastings laid the foundations for what was to become an entire body of research for MCMC.

Monte Carlo Markov Chains are the intersection of both Monte Carlo methods and Markov Chain. MCMC allows the user to draw on the favorable properties of both statistical methods. The Monte Carlo portion of MCMC is quite similar to what has been previously discussed. Random samples are taken from some distribution (normal, exponential, etc.) or process. The user then calculates the desired summary statistics or metrics for those samples of observations. The Markov Chain property of MCMC follows that each of these random samples taken in the Monte Carlo portion of MCMC are steps, or transitions, generated by some system or process [9]. Each step in the markov chain would be an independent monte carlo simulation. One state of the chain might indicate to draw a sample from one type of distribution, another state would indicate some other distribution, and so on for the total number of states. The probability of each state occurring is defined in the transition matrix. As with the memoryless property of Markov Chains, the state of that system in the next step or transitional state only depends on the current state. New samples taken in the Monte Carlo portion of the MCMC do not depend on samples taken in the past.

As with all methods in statistics, MCMC come with their own sets of advantages and disadvantages. As will be demonstrated in our later tutorial and example, MCMC methods are extremely effective at handling highly complex datasets that have many dimensions and dependencies [9]. Referring to our weather example from before, suppose the weatherman tacked on temperature, wind speed, humidity, and air pressure as additional dimensions in that example. The Markov chain would be able to handle calculating transitional probabilities for all those states. A major disadvantage that comes with MCMC is the *convergence* problem highlighted in many of the references here. MCMC relies on an initial condition, or ‘guess’, that is input from the user. If this initial condition is extremely wrong and highly unlikely to come from the target distribution, the algorithm will take an extremely long time to converge (if it ever does) to the target distribution [6,9]. The user must be relatively confident that the initial condition is within the outcome space of the target distribution to ensure convergence.

Theory & Application: Monte Carlo Markov Chains in Bond Pricing

Treasury bonds are important financial instruments frequently used to hedge risk, debt, and manage a bank's balance sheet. Managing the risk associated with their price and movements in their prices under a range of financial scenarios are critical for sound financial risk management. In 2023, it was broadly reported that Bank of America had a portfolio of unrealized bond losses that measured over $100B [10]. Following the COVID-19 pandemic, the large influx of consumer deposits from the CARES act stimulus checks meant banks were flush with cash. Bank of America decided to use that influx of cash to begin the purchase of longer-dated treasury bonds. This broad mismanagement of its bond portfolio involving a mismatch of price, duration, and interest rates has led to the bank being exposed to these significant amounts of losses. This section will walk through how MCMC could be applied to the pricing of a bond portfolio, provide a simple example of its implementation, and analyze its output.

Monte Carlo Method:

One of the many applications of Monte Carlo methods in finance is to price a treasury bond. Bonds are instruments issued by governments (both state and federal) and corporations to finance spending or manage their debt burdens. The price of a bond is directly related to changes in interest rates. Interest rates fluctuate because of Federal Reserve Policy, economic conditions, political events, or geographic events. The price of a bond can be generally calculated as follows:

$$P = \sum_{t=1}^{n} \left(\frac{C}{(1+r)^t} + \frac{F}{(1+r)^n} \right)$$

Where P is the price of the bond, C is the coupon payment, r is the interest rate (yield), F is the face value of the bond, and n is the number of periods until maturity [11]. The variables C, F, n, and t are all constants set by the users and parameters of the bond. The component r in the above formula is uncertain and can fluctuate with time due to extraneous events. A property inherent to bonds is that interest rate and price is inversely related.

A Monte Carlo simulation for pricing a bond would be primarily driven by generating a distribution of simulated values [3] for the interest rate variable[1]. For the bond example, we will use a base case, high volatility[2] case, and low volatility case to demonstrate how pricing would change if the distribution of interest rates shifted. Our Low volatility case would be represented by a normal distribution with a mean of 2% and a standard deviation of 0.075%. The base volatility case is a normal distribution with a mean of 2% and standard deviation of 0.1%. The high volatility case is a normal distribution with a mean of 2.0% and standard deviation of 0.2%. Interest rates can follow a variety of distributions on a wide array of products, so users should be sure to properly analyze and define their sampling distributions prior to input into the simulation.

---

[1] For simplicity, only interest rates will be simulated. Coupon rates can also be simulated in a similar fashion but are fixed in this example. Assumes one coupon a year, a face value of $1,000, a 5-year term, and a fixed coupon rate of 5%.

[2] Volatility is a measure of fluctuation of bond interest rates. In this example, it would be the standard deviation of the simulated interest rates

Figure 1 displays the sample distribution of those interest rates. The high volatility case has a wider spread of possible interest rates compared to the base case and the lower volatility has a much tighter spread of possible interest rates compared to the base case. These simulated interest rates from figure 1 can now be used to generate resulting prices of bonds. Figure 2 shows the resulting distribution of bond prices in each volatility case.

Figure 1: Interest Rate Volatility Distribution

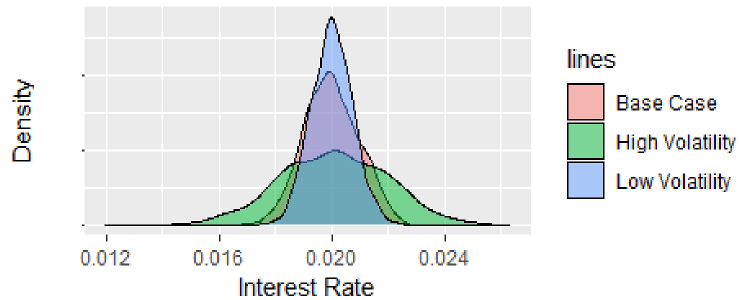
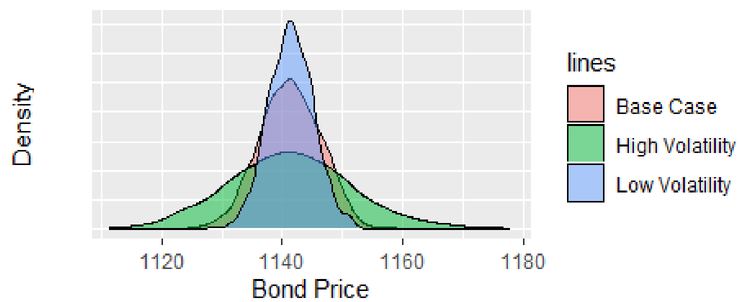


Figure 2: Resulting Bond Price Distribution



Markov Chain Method:

Markov chains are a stochastic process in which the next state of a modeled system is only dependent on the current state of the system and is independent of anything that occurred before that. The probability that the current state transitions into *n* number of next states is determined and set by a *transition matrix*. The transition matrix is an input that must be defined by the user. Each output state (or 'next state') of the transition matrix can also be defined by the user [6,7]. Most often, the probability for each transition state is determined by a *prior distribution* that is observed and determined mathematically.

Each state of the matrix can be described as each volatility condition for the random variable of interest in the bond example described previously. If sampling from an actual treasury bond, these transition probabilities into each different state of the system could be empirically calculated - but since the example uses a fictitious example bond, we can set the transition probabilities manually. The transition matrix will look like the following:

Table 1: Transition Matrix Layout

		Volatility State (Next State ~ Y)		
		High	**Base**	**Low**
Volatility State (Current state ~ X)	**High**	Pr (Y = High | X = High)	Pr (Y = Base | X = High)	Pr (Y = Low | X = High)
	Base	Pr (Y = High | X = Base)	Pr (Y = Base | X = Base)	Pr (Y = Low | X = Base)
	Low	Pr (Y = High | X = Low)	Pr (Y = Base | X = Low)	Pr (Y = Low | X = Low)

Every cell in Table 1 represents the probability that the current state, represented by *X*, transitions to a next state, represented by *Y*. Aligned with the memoryless property of Markov chains, what state the system transitions into next is only dependent on the current state of the system None of the probabilities in Table 1 are dependent on any realization of the system older than the current state of *X*.

The sum across each row of probabilities should equal to one, incumbent on the law of total probabilities [6]. If the current state of the system is Base, it can only transition into three other states - High, Base, and Low. There are no other transitional states, representing the total sample space of outcomes. Therefore, the sum of the probabilities of each row is equal to one.

For our example presented here, we'll assume that the transition matrix looks like Table 2 presented below:

Table 2: Hypothetical Bond Price Transition Matrix				
	Volatility State (Next State ~ Y)			
Volatility State (Current state ~ X)		**High**	**Base**	**Low**
	High	0.2	0.5	0.3
	Base	0.4	0.5	0.1
	Low	0.1	0.6	0.3

After implementing Table 2 into a Markov chain utilizing the markovchains package in R and calculating n = 1000 sequential states, the vector of outputs will look something like the following:

Interest Rate Volatility State Markov Chain = "Base", "Low", "Low", "High", "Base"......

Which produces the following frequency table across all 1000 realizations of the Markov chain:

State	**Frequency**
Base	510 (51.0%)
High	198 (19.8%)
Low	292 (29.2%)

'Base Volatility' occurs 51% of the time, 'High Volatility' occurs 19.8% of the time, and 'Low volatility' occurs 29.2% of the time. The sum equals the total number of steps in the chain (*n = 1000).*

One important additional facet of a Markov chain is the *initial condition* of a Markov chain [6,7]. The initial condition is the state of a system at *n=0*, the state the Markov chain is initialized in. Changing the initial state of the Markov chain can impact the resulting distribution of outcomes, presented in the below table. For some applications, the initial condition can be significant and should be thoroughly evaluated for robustness. For our bond example, the initial condition used will be the 'Base' condition.

State	Initial Condition = Base	Initial Condition = High	Initial Condition = Low
Base	510	496	524
High	198	210	210
Low	292	294	266

Monte Carlo Markov Chains in Bond Pricing:

Monte Carlo Markov Chains combine both of the previously introduced Monte Carlo methods and Markov chain methods to produce a randomized stochastic system that samples from distributions as it progresses along a sequential system. Continuing with our bond pricing example, each step of the Markov chain specifies a volatility state that determines the distribution of interest rates the system will sample from. When the Markov chain specifies a *Base* state the system will sample from the *Base* volatility distribution (and the same for the other interest rate volatility distributions). Figure 3 illustrates what this process will look like:

Step 1	Determine Initial State
Step 2	Sample relevant distribution based on result of Markov chain
Step 3	Utilize output of the system to calculate resulting bond prices
Step 4	Determine new system state using Markov chain
Step 5	Repeat Steps 2-4 for number of steps in the chain

Implementing figure 3 to our bond price example is illustrated below:

Step	*i0*	*i 1*	*i 2*
Markov Chain State	*Base*	*High*	*Low*
Sampled Distribution	*Base Volatility*	*High Volatility*	*Low Volatility*
Distribution	*Norm(0.02, 0.001)*	*Norm(0.02, 0.002)*	*Norm(0.02, 0.00075)*
Interest Rate	2.08%	2.20%	2.03%
Resulting Bond Price	$1,137	$1,131	$1,140
Next state generated by chain	High	Low	Base

The system starts in the initial condition. For the initial condition *i0*, the system is in base volatility. Base volatility requires sampling from a normal distribution with a mean of 2% and a standard deviation of 0.1%. This run of the system generated an interest rate of 2.08%, which produces a bond price of $1,137. The markov chain generated a volatility state of 'high' for the next iteration, so the calculation repeats the prior steps utilizing this state. This process repeats for *n* number of steps per chain. The user can also generate *j*-number of chains to increase its accuracy [7].

Running 100 (*j=100*) samples of our markov chain based on the transition matrix in Table 2 and the initial state of base produces an interest rate path presented in Figure 3. At each system step, an interest rate is generated by sampling from the relevant distribution. The pattern of the MCMC centered around 2%, equivalent to the mean of each of the three state distributions. Due to the randomness of the transition matrix and the random samples drawn at each step, slight drifts above and below the mean are observed.

To calculate the terminal bond price for each of the *j* MCMC iterations, the 100th step for each of the *j* chains is used. Figure 4 and Figure 5 both present the resulting distribution of both the resulting interest rate that was used as well as the resulting bond based on the interest rate that was used. A bimodal distribution can be observed for both distributions. The most common value appears around 2.0%, which is the mean of the three interest rate distributions used, but there is another common value around 1.7%. This can be attributed to both the randomness in the distributions used, but also the randomness of the Markov chain transition matrix. While arbitrary transition probabilities were used in our example, financial economists can adjust transition probabilities based on their assessment of external economic and market conditions to generate a more accurate result.

Figure 3: Rate Produced at Markov Chain Step

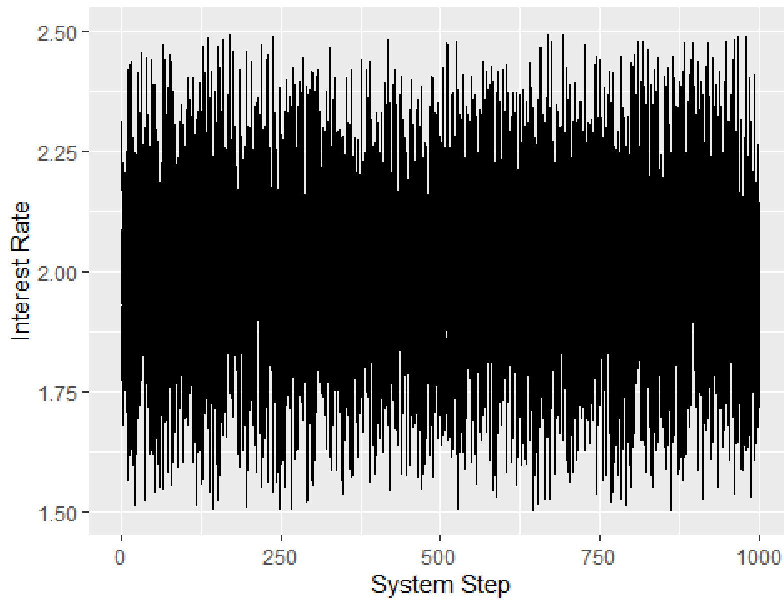


Figure 4: Terminal Interest Rate for MCMC Iterations

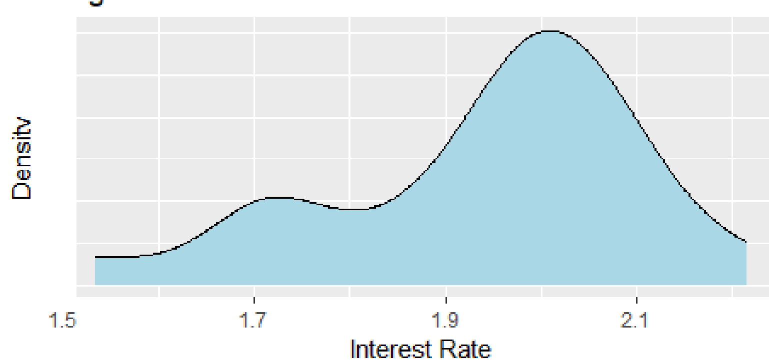
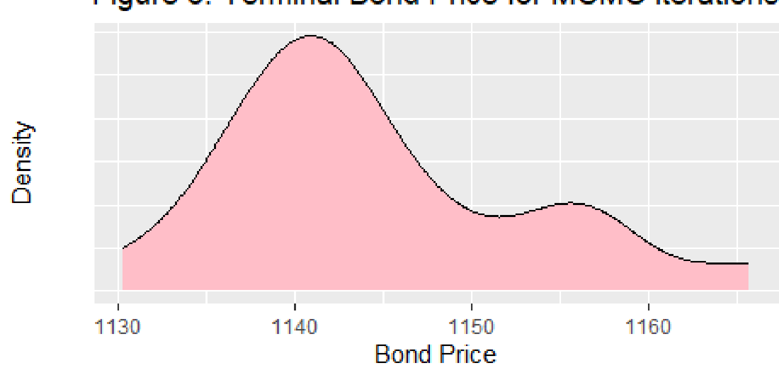


Figure 5: Terminal Bond Price for MCMC Iterations



Conclusion:

Monte Carlo Markov chains are evolving into a useful tool in finance to predict the terminal price of an asset based on underlying transition probabilities and distributions. This work briefly surveyed the history of MCMC and the origin of its constituent elements (Monte Carlo simulations and Markov chains). Basic examples of each were presented in each specific context as well as the advantages and disadvantages of each. Then, we stepped through a simple example applying the combined approach of Monte carlo markov chains to simulating the price of a bond dependent on an interest rate that would be sampled from an MCMC. We created volatility distributions based on pre-selected market dispositions. We also generated a sample transition matrix to model movements between volatility states. The report then applied both the sample volatility distributions and transition matrix via a MCMC to produce a resulting interest rate distribution and bond price. We concluded by showing what the resulting distributions of both price and interest rate look like.

An immediate improvement to the methods presented here would be to introduce additional random variables to model. An important factor in bond pricing is both the coupon rate as well as the federal open market window rate. For this example, we assumed that the coupon rate is constant and did not include a factor for the open market rate. Open market rates are extremely important in the pricing of a bond and are rarely constant over time [11], so introducing that element to our simulation could only improve the accuracy of our results. Additionally, building our simulation on top of an actual bond instead of a fictitious bond used in this example would also improve our results. For ease of implementation and to apply a focus on theory, this paper did not apply our methods to an actual financial security. An obvious next step would be to select a bond from the listing on the exchange and apply these methods to that instrument. Doing so would allow a user the benefit of applying this framework to an actual security as well as to be able to better define a transition matrix instead of the fixed parameters supplied here.

References:

[1] Crotty, J. (2009). Structural causes of the global financial crisis: a critical assessment of the new financial architecture'. Cambridge Journal of Economics, Vol 33 Issue 4.

[2] Federal Reserve. (n.d.). *Comprehensive capital analysis and review: Questions and answers*. Retrieved July 12, 2024, from https://www.federalreserve.gov/publications/comprehensive-capital-analysis-and-review-questions-and-answers.htm

[3] Harmra, G. MacLehose, R. Richardson, D. (2013). Markov Chain Monte Carlo: an introduction for epidemiologists.

[4] Harrison, R. (2011). Introduction to Monte Carlo Simulation. AIP Conf. Proc.

[5] Spiceworks. (n.d.). *What is a Monte Carlo simulation?* Retrieved July 13, 2024, from https://www.spiceworks.com/tech/tech-general/articles/what-is-a-monte-carlo-simulation/

[6] Mahfuz, F. (2021). Markov Chains and their applications. Math Theses. University of Texas at Tyler.

[7] Harmra, G. MacLehose, R. Richardson, D. (2013). Markov Chain Monte Carlo: an introduction for epidemiologists.

[8] Coppi, P. (2008) A brief history of Markov Chain Monte Carlo: Subjective recollections from incomplete data. Retrieved from: http://www.astro.yale.edu/coppi/astro520/mcmc/0808.2902v4_mcmc_history.pdf

[9] Ravenzwaaij, D., Cassey, P., & Brown, S. (2016). A simple introduction to Markov Chain Monte-Carlo Sampling. Psychon Bull Rev.

[10] Barron's. (n.d.). Bank of America stock, bond losses. *Barron's*. Retrieved July 12, 2024, from https://www.barrons.com/articles/bank-of-america-stock-bond-losses-97bf300e

[11] Investopedia. (n.d.). Bond valuation. *Investopedia*. Retrieved July 12, 2024, from https://www.investopedia.com/terms/b/bond-valuation.asp