OPTION PRICING AND CVA CALCULATIONS USING THE MONTE CARLO-TREE (MC-TREE) METHOD

Yen Thuan TRINH
School of Mathematical Sciences
University College Cork
Ireland
trinhthuanyen2017@gmail.com

Bernard HANZON, Chair of Mathematics
School of Mathematical Sciences
University College Cork
Ireland
b.hanzon@ucc.ie

ABSTRACT

The binomial tree method and the Monte Carlo (MC) method are popular methods for solving option pricing problems. There is need for faster and more accurate option price calculations. We introduce a new method, the MC-Tree method, that combines the MC method with the well-known recombining binomial tree based on Pascal’s triangle for pricing single asset options. Our approach uses a mixing distribution on the tree, for which we obtain the corresponding compound distribution on the tree outcome. As well known in the literature, the standard Gaussian distribution is the distribution with the maximal entropy among distributions with zero mean and unit variance. The compound density that we obtain is not exactly equal the Gaussian density, but has very high entropy. Also we introduce techniques to correct for the deviation from the Gaussian. Based on these techniques we develop an algorithm for calculation of the Credit Valuation Adjustment (CVA) on
an American put option, a challenge in computation due to the complexity of the American options and modelling CVA. We also present numerical results. Based on our these, the MC-Tree method is more accurate than the well-known Least Square Monte Carlo method (LSM) for American option valuation proposed by Longstaff Schwartz (2001) at the same numbers of simulations. The MC-Tree method performs better than other methods: Cox, Ross Rubenstein (CRR) and Jarrow-Rudd(JR) in terms of accuracy, using the same tree depth. Also, the MC-Tree method with the distribution correction technique dramatically improved the accuracy, resulting in the practically exact solutions, compared to analytical solutions, at the tree depth N=50 or 100 and MC-drawings M=100000. The bias-correction technique makes the resulting tree model complete in the sense of financial mathematics and obtains the risk-neutral probability.

**Keywords** European Options, American Options, Binomial Trees, Monte Carlo Method, Counterparty Credit Risk, Credit Valuation Adjustment.

1 **Introduction**

The pioneering work of Bachelier (1900) is now seen as the forerunner of what would become a massive usage of mathematical models in finance since the last quarter of the 20th century. There is a demand for fast and accurate methods for pricing options when more exotic financial instruments are developed and traded on the market over the years. Black and Scholes (see [3]) derived the Black-Scholes equation to price derivatives on a single asset. Subsequently analytical solutions for quite a number of derivatives have been derived, using either the Black-Scholes partial differential equation or the discounted risk-neutral expectation method, but it is too difficult to solve the equation analytically for an
arbitrary derivative. Based on the Black-Scholes model, many methods for option valuation have been developed, but there is still room for improvement. Our contribution is to introduce a new computational approach to find the value of financial derivatives with arbitrary boundary conditions and generalization to the case of American options and other classes of options such as Barrier options.

Boyle [4] introduced the Monte Carlo approach to option pricing, and it is still very popular because of its flexibility to approximate all kinds of option prices. Monte Carlo simulation has been extended in pricing American options with popular methods such as the Stochastic Mesh [11], the Least Square Monte Carlo [23], and the State-Space partition [33], and so on. Longstaff and Schwartz in [23] introduced a least square regression method (LSM) to approximate the prices of American options. Much research has been done subsequently to give the analysis and the convergence of the LSM.

Lattice-type models for pricing options were implemented in many works ( [5]; [6]). Various choices for tree parameters lead to different existing binomial models. The first proposed formulation of the binomial tree in financial mathematics was the research by Cox, Ross, and Rubinstein [14], providing a simplified discrete approach to option pricing on one asset using the recombining binomial trees based on Pascal’s triangle. The authors in this article proposed to take \( u = 1/d \), where \( u, d \) are upward and downward movements. The work is still very popular today. One benefit of the model is that it exhibits a unique risk-neutral probability. Sierag and Hanzon [30] extended this to multi-asset option pricing using recombining multinomial trees based on Pascal’s simplex. The second popular model was the research by Jarrow and Rudd [19], also known as the equal-probability model. One drawback is that the choice of equal probability seems to be unrealistic. Another drawback is that the model is no longer a risk-neutral model although it matches the risk-neutral continuous-time model in the limit for the time-step length going to zero. Authors in ( [14], [19]) matched the first two moments to the risk-neutral continuous model, leading
to a system of two equations for three unknowns. Tian’s approach [32] was to equate the third moment to handle the issue of this free variable. Leisen and Reimer [20] presented the most different choice of the model parameters $u, d$ and $p$ among the approaches mentioned above. They devoted attention to improving the convergence rate and smoothness when approximating the Gaussian distribution.

The main advantage of classical methods such as the Tree or Lattice method is its simplicity for implementation in option pricing. A higher tree depth is required to get good precision. As binomial trees are recombining they greatly reduce the number of nodes compared to general binary trees. As a result, the corresponding computational cost is reduced.

Options and other financial derivatives form an important section of the financial markets. The majority of exchange-traded options on a single asset tend to be American style, while options on indexes are European. After the financial crisis of 2007-2008, there is a requirement for a significant change in financial modelling and risk management, which is reflected in the Basel Accords-Basel I, Basel II, and Basel III for the calculations of the required capital issued by the BIS (Bank of International Settlements). The financial regulators required banks to hold an amount of capital to capture the credit risk in portfolios. The BIS published Basel III to work along with Basel II in response to the deficiencies in banking regulation in Basel I. Basel III presented a new measure, namely Credit Valuation Adjustment to capture Counterparty Credit Risk (CCR) [2]. As a result there is an increased interest in CVA. This is the motivation for us to develop one algorithm for the calculation of the Credit Value Adjustment (CVA) on an American option, using our MC-Tree technique. If the exposure profile and the credit quality of the counterparty are positively or negatively related, so-called right-way risk (RWR) or wrong-way risk (WWR) occurs. Otherwise, one speaks about unilateral CVA.

In this article, unilateral CVA will be computed for an American put option based on standard assumptions, using the MC-Tree method.
Various existing versions of binomial models use different choices for parameters. We aim to improve the accuracy and speed of the binomial method by applying Monte Carlo simulation on these parameters. The idea is to generate the tree directions and probabilities through a parameter. This parameter is drawn from a probability density, called the mixing density. A formula will be provided for the resulting compound density that is then generated by the tree. The goal is to find a mixing density such that the corresponding compound density is close to a Gaussian density, working with the additive representation of the tree. It does not seem to be possible to find a mixing density for which the compound density is exactly Gaussian. We succeed in specifying a class of mixing densities for which the compound density is rational (and hence smooth). The standard Gaussian density can be approximated by rational densities as follows from the following well-known limit:

\[
\lim_{{N \to \infty}} \frac{1}{{(1 + \frac{x^2/2}{N})^N}} = e^{-\frac{x^2}{2}}.
\]

Therefore, it is not unreasonable to construct rational compound densities to approximate the standard Gaussian density function. It can be proved that the standard Gaussian distribution can not be obtained in a 1-step tree. The reason is that standard Gaussian can be obtained on the negative half-line, which means that the mixing density can be computed so that the compound density is equal to the standard Gaussian density on the negative half-line. However, this requirement fully determines the mixing density, and this mixing density produces a compound density on the positive half-line that is not a standard Gaussian at all.

To get good approximations of the Gaussian distribution, we try to find mixing densities for which the entropy of the compound density is high. The motivation is that our tree construction forces it to have a distribution on the end-nodes at step N with zero mean and variance equal to N. This will also hold for the compound density of any MC-Tree. If \( \tau \)
denotes a parameter characterizing a 1-step tree and X the random variable resulting from
the MC-Tree procedure at time step N, then
\[ E_{X|\tau}(X) = 0, \quad E_{X|\tau}(X^2) = N. \]
It implies that
\[ E_X(X) = E_\tau(E_{X|\tau}(X | \tau)) = 0, \quad E_X(X^2) = E_\tau(E_{X|\tau}(X^2 | \tau)) = N. \]
It is well-known from the literature that the standard Gaussian distribution \( \mathcal{N}(0, 1) \) with the
pdf \( f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \) maximizes the entropy integral
\[ \text{Ent}(f) := -\int_{-\infty}^{\infty} f(x) \log(f(x)) \, dx, \]
subject to the constraints
\[ \int_{-\infty}^{\infty} f(x) \, dx = 1, \quad \int_{-\infty}^{\infty} x f(x) \, dx = 0, \quad \int_{-\infty}^{\infty} x^2 f(x) \, dx = 1 \]
and more generally a Gaussian density with given mean and variance maximizes the entropy among all densities with that
same mean and variance.

### 2.1 Parameters

The multiplicative binomial tree has the probability \( p_1 > 0 \) of moving "down" to \( S^n e^d \) and
\( p_2 > 0 \) of moving "up" to \( S^n e^u \), where \( p_1 + p_2 = 1 \) and \( S^n > 0 \). Here, \( u > d \); this is actually
all that is required, and \( u > 0 \) and \( d < 0 \) is not required. So for \( n = 0, 1, \ldots \), we have

\[
S^{(n+1)\delta t} = \begin{cases} 
S^n e^u & \text{w.p. } p_2 \\
S^n e^d & \text{w.p. } p_1.
\end{cases}
\]

Equivalently, we have the following additive tree.

\[
\log(S^{(n+1)\delta t}) = \begin{cases} 
\log(S^n) + u & \text{w.p. } p_2 \\
\log(S^n) + d & \text{w.p. } p_1.
\end{cases}
\]

As the mean and variance can always be adapted using an affine transformation, we will
first consider the case in which mean=0 and variance=1 for each time-step. This implies
\[ p_2 + p_1 = 1, \]
\[ p_2u + p_1d = 0, \]
\[ p_2u^2 + p_1d^2 = 1, \]
\[ p_2 > 0, \quad p_1 > 0. \]

The family of all solutions can be parametrized by an angle \( \theta \), with \( 0 < \theta < \frac{\pi}{2} \), as follows:

- \( p_1^{1/2} = \cos(\theta) \),
- \( p_2^{1/2} = \sin(\theta) \),
- \( u = \sqrt{\frac{p_1}{p_2}} \),
- \( d = -\sqrt{\frac{p_2}{p_1}} \).

For a geometric interpretation of the angle \( \theta \), we refer to Sierag and Hanzon \([30]\). Hence, we now consider \( \theta \) to be a random variable with distribution function \( P_m \), supported on \( (0, \pi/2) \), providing us with a mixing distribution on the tree. We can summarize this schematically as follows:

\[ \log(S^N) \mid \theta \sim \text{Binomial}, \quad \theta \sim P_m. \]

Here by "Binomial" we mean the distribution of \( \log(S^N) \) at the final nodes of the tree. Combining the binomial distribution on the log-asset-prices with the distribution on \( \theta \) will result in a compound density for \( \log(S^N) \). We will say more about how the compound density can be calculated in the next section.

The proposed approach is now to compute an expected value of a payoff function defined on \( \log(S^N) \), with respect to the compound density by (1) assembling, say, \( M \) drawings of the variable \( \theta \) and (2) for each \( \theta \) to "run the tree" to obtain an approximation to the expected value of the payoff, and (3) to compute the average and standard deviation of the tree-outcomes to obtain a Monte Carlo estimation of the expected payoff, as well as a confidence interval. Informally we refer to this procedure as "shaking the tree". In terms of Monte Carlo theory, this method falls under the category "variance reduction by conditioning" \([7]\).
The idea is that because the tree outcomes will already be very close to the true value (especially for deeper trees), the Monte Carlo outcomes will be very accurate.

2.2 Preliminaries to derivation of the general compound density formula

Let \( N \in \mathbb{N} \). Define

\[
x_{N,k} = -(N - k)\tau + \frac{k}{\tau}, \quad k = 0, 1, 2, \ldots, N.
\]

Note that

\[
\text{Range of } x_{N,k} = \begin{cases} (-\infty, 0) & \text{if } k = 0, \\ \mathbb{R} & \text{if } k = 1, 2, \ldots, N - 1, \\ (0, \infty) & \text{if } k = N. \end{cases}
\]

As the derivative \( x'_{N,k}(\tau) = -(N - k) - \frac{k}{\tau^2} < 0 \) for all \( \tau \in (0, \infty) \), it follows that \( x_{N,k} \) is monotonically decreasing for each \( N \in \mathbb{N} \) and \( k \in \{0, 1, \ldots, N\} \). Hence, \( x_{N,k}(\tau) \) has an inverse function \( \tau_k(x) \) with domain \((-\infty, 0)\) if \( k = 0 \), \( \mathbb{R} \) if \( k = 1, \ldots, N - 1 \) and \((0, \infty)\) if \( k = N \); and range \((0, \infty)\) in all cases.

As can easily be verified, one has

\[
\tau_k(x) = \frac{-x + \sqrt{x^2 + 4k(N - k)}}{2(N - k)} \quad \text{if } k \neq 0, N,
\]

\[
\tau_0(x) = -\frac{x}{N}, \quad x \in (-\infty, 0),
\]

\[
\tau_N(x) = \frac{N}{x}, \quad x \in (0, \infty).
\]

Note that \( \tau_k(x)\tau_{N-k}(-x) = 1 \) holds for all \( x \) for which the left hand side is defined. For later reference, we also define \( y_k(x) = \sqrt{x^2 + 4k(N - k)} \), \( k = 1, 2, \ldots, N - 1 \). The variable \( y_k \) can also be expressed in terms of \( \tau_k \), as follows.

\[
y_k(x) = (N - k)\tau_k + \frac{1}{\tau_k} = (N - k)\tau_k(x) + k\tau_{N-k}(-x).
\]

Now, for an additive binomial tree with \( d = -\tau \) with the probability \( \frac{1}{1+\tau^2} \) and \( u = \frac{1}{\tau} \) with the probability \( \frac{\tau^2}{1+\tau^2} \), the probability distribution of the values at the nodes at the tree depth...
N can be expressed as:

\[ X_{N|\tau} = \begin{cases} 
  x_{N,0}(\tau) & \text{with prob } g_0(\tau) := \frac{1}{(1+\tau^2)^N}, \\
  x_{N,k}(\tau) & \text{with prob } g_k(\tau) := \binom{N}{k} \frac{\tau^k}{(1+\tau^2)^N}, \quad k = 1, N - 1, \\
  x_{N,N}(\tau) & \text{with prob } g_N(\tau) := \frac{(\tau^2)^N}{(1+\tau^2)^N}.
\]

Note that \( X_{N|\tau} \) is the sum of \( N \) stochastically independent copies of the random variable \( X_{1|\tau} \), and hence it has mean \( \mathbb{E}[X_{N|\tau}] = 0 \), and the variance at \( \mathbb{E}[(X_{N|\tau})^2] = N \). So \( \frac{1}{\sqrt{N}} X_{N|\tau} \) is a random variable with mean zero and variance one.

Application of the Central Limit Theorem tells us that the Cumulative Distribution Function (CDF) of \( \frac{1}{\sqrt{N}} X_{N|\tau} \) converges to the CDF of a Standard Gaussian random variable for \( N \to \infty \) (and \( \tau > 0 \) fixed). The CDF of \( X_{N|\tau} \) can be described as follows.

\[
F(x | \tau) = P(X_{N|\tau} \leq x | \tau) = g_0(\tau) \mathbb{1}_{\{x \geq x_{N,0}(\tau) \ & \ & x < 0\}} + \sum_{k=1}^{N-1} g_k(\tau) \mathbb{1}_{\{x \geq x_{N,k}(\tau)\}} + g_N(\tau) \mathbb{1}_{\{x \geq x_{N,N}(\tau) > 0\}}.
\]

If we now consider the tree parameter \( \tau \) as random with the probability density function \( p_m(\tau) \) (we will call \( p_m(\tau) \) the "mixing density") then the resulting compound CDF of X will be \( Q(x) := \int_0^\infty F(x | t)p_m(t)dt \).

### 2.3 General Compound Density Formula

**Theorem 1.** The compound probability density function \( q(x) = Q'(x) \) of \( X \) satisfies the following formula

\[
q(x) = \sum_{k=0}^{N} C_k(x),
\]

where:

- \( C_0(x) = \frac{1}{(1+\tau_0^2)^N} p_m(\tau_0) \frac{1}{N} \mathbb{1}_{\{x \leq 0\}} \)

- \( C_k(x) = \binom{N}{k} \frac{(\tau_k^2)^k}{(1+\tau_k^2)^N} p_m(\tau_k) \frac{\tau_k}{y_k}, \quad k = 1, N - 1. \)
\[ C_N(x) = \frac{(\tau_N^2)^N}{(1 + \tau_N^2)^N} p_m(\tau_N) \frac{N}{x^2} \mathbb{1}_{\{x > 0\}}. \]

**Proof.** Fix \( N \). As for each \( k \in \{0, \ldots, N\} \), the function \( x_{N,k}(\tau) \) is monotonically decreasing with inverse \( \tau_k(x) \), we can write

\[
Q(x) = \int_0^\infty F(x \mid \tau) p_m(\tau) d\tau = \mathbb{1}_{\{x < 0\}} \int_{\{\tau \geq \tau_0(x) \& \, x < 0\}} g_0(\tau) p_m(\tau) d\tau + \sum_{k=1}^{N-1} \int_{\{\tau \geq \tau_k(x)\}} g_k(\tau) p_m(\tau) d\tau + \mathbb{1}_{\{x > 0\}} \int_{\{\tau \geq \tau_N(x) \& \, x > 0\}} g_N(\tau) p_m(\tau) d\tau.
\]

For \( k = 1, \ldots, N - 1 \), we have

\[
1 = \frac{d}{dx} x_{N,k}(\tau_k(x)) = \{- (N - k) - \frac{k}{\tau_k^2}\} \tau'_k = \{- ((N - k)\tau_k + k\frac{1}{\tau_k^2})\} \frac{\tau'_k}{\tau_k} = -\frac{y_k}{\tau_k} \tau'_k.
\]

Therefore,

\[
\tau'_k(x) = -\frac{\tau_k}{y_k}, \quad k = 1, \ldots, N - 1.
\]

Recall \( \tau_0(x) = -\frac{N}{x} \Rightarrow \tau'_0(x) = -\frac{1}{x}, \ x < 0 \), and \( \tau_N(x) = \frac{N}{x} \Rightarrow \tau'_N(x) = -\frac{N}{x^2}, \ x > 0 \). Now taking the derivative of \( Q(x) \), we obtain

\[
q(x) = \mathbb{1}_{\{x < 0\}} g_0(\tau_0(x)) p_m(\tau_0(x)) \frac{1}{N} + \sum_{k=1}^{N-1} g_k(\tau_k(x)) p_m(\tau_k(x)) \frac{\tau'_k(x)}{y_k(x)} + \mathbb{1}_{\{x > 0\}} g_N(\tau_N(x)) p_m(\tau_N(x)) \frac{N}{x^2} = C_0(x) + \sum_{k=1}^{N-1} C_k(x) + C_N(x).
\]

\[
\square
\]

### 2.4 Mixing Density

As is well-known, due to the convexity of the function \( g(y) = y \log(y) \), if \( f(x) \) is a pdf on \( \mathbb{R} \) then \( \frac{f(x) + f(-x)}{2} \) has entropy at least as high as \( f(x) \). As we are looking for compound densities with high entropy, the consequence of this is that we can restrict our search to mixing densities that produce an even compound density function. In terms of the mixing distribution, this translates into considering mixing densities which are invariant under a permutation of the two axes in the binomial tree. So we use mixing probabilities on the
recombining binomial tree such that symmetric paths have the same probability. A relatively simple class of mixing densities satisfying this invariance is given, in terms of the parameter $p_1$, by
\[
\frac{1}{2} c_m (p_1^{1/2} p_2^{1/2})^{m-2} dp_1, \ m \in \mathbb{N},
\]
where $p_1 > 0$, $p_2 > 0$, $p_1 + p_2 = 1$ and $c_m$ is a normalization constant. Recalling $p_1 = \cos^2(\theta)$, $p_2 = \sin^2(\theta)$, the transformation $\tau = \tan(\theta)$ leads to $p_1 = \frac{1}{1+\tau^2}$, $p_2 = \frac{\tau^2}{1+\tau^2}$, so
\[
\frac{1}{2} c_m (p_1^{1/2} p_2^{1/2})^{m-2} dp_1 = \frac{1}{2} c_m \left(\frac{1}{1+\tau^2}\right)^{(m-2)/2} \frac{\tau^2}{1+\tau^2}^{(m-2)/2} \left|d\left(\frac{1}{1+\tau^2}\right)\right| = c_m \frac{\tau^{m-1}}{(1+\tau^2)^m} d\tau,
\]
where the constant $c_m$ is given by
\[
c_m = \frac{1}{\int_0^\infty \frac{\tau^{m-1}}{(1+\tau^2)^m} \ d\tau}.
\]
In terms of the parameter $\theta$ we obtain a third representation of these mixing densities:
\[
\frac{1}{2} c_m (p_1^{1/2} p_2^{1/2})^{m-2} dp_1 = c_m (\cos(\theta) \sin(\theta))^{m-1} d\theta.
\]
The idea is now to apply the MC technique and draw $\tau$ from this probability distribution on $(0, \infty)$. We will also make use of the transformation $\tau = \tan(\theta) \iff \theta = \arctan(\tau)$, $\theta \in (0, \pi/2)$ regularly. Drawing $\tau$ can then be replaced by drawing $\theta$ and using $\tau = \tan(\theta)$.

### 2.5 Monte Carlo Drawing

In order to carry out the Monte Carlo simulations we need to be able to draw independent samples from the mixing distribution. A general technique in case the cumulative distribution function (CDF) is available is to draw random samples from the uniform distribution on the interval $[0, 1]$ and to use the inverse function of the CDF to obtain the desired samples. Note that as our mixing densities are everywhere positive that the inverse of its CDF exists, and given any drawing from the uniform distribution, the corresponding sample from the mixing distribution can be found, for instance, by a bisection method. Therefore what remains is to find the CDF of our mixing distributions. One way to do that is to work out the CDF of the mixing density in terms of the angle $\theta$. This can be done as follows:
Note that

\[ p_m(\theta) = c_m \cos(\theta)^{m-1} \sin(\theta)^{m-1} = c_m \left( \frac{e^{i\theta} + e^{-i\theta}}{2} \right)^{m-1} \left( \frac{e^{i\theta} - e^{-i\theta}}{2i} \right)^{m-1} = \]

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m-1} (e^{i2\theta} - e^{-i2\theta})^{m-1} \right] = \]

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m-1} e^{-i2\theta(m-1)}(e^{i4\theta} - 1)^{m-1} \right] = \]

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m-1} e^{-i2\theta(m-1)} \sum_{s=0}^{m-1} \left( \frac{m-1}{s} \right) e^{i4s\theta} (-1)^{m-1-s} \right] = \]

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m-1} \sum_{s=0}^{m-1} \left( \frac{m-1}{s} \right) e^{i(4s-2(m-1))\theta} (-1)^{m-1-s} \right]. \]

• In case \( m \) is odd, this has the following primitive function

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m-1} \sum_{s=0, s \neq \frac{m-1}{2}}^{m-1} \left( \frac{m-1}{s} \right) \frac{(-1)^{m-1-s}}{i(4s-2(m-1))} e^{i(4s-2(m-1))\theta} + \right. \]

\[ (-i)^{m-1} \left( \frac{m-1}{(m-1)/2} \right) \theta (-1)^{(m-1)/2} + \tilde{C}_m = \]

\[ c_m 2^{-2(m-1)} \ Re \left[ (-i)^{m} \sum_{s=0, s \neq \frac{m-1}{2}}^{m-1} \left( \frac{m-1}{s} \right) \frac{(-1)^{m-1-s}}{2s -(m-1)} e^{i(4s-2(m-1))\theta} + \right. \]

\[ (-1)^{(m-1)/2} \left( \frac{m-1}{(m-1)/2} \right) \theta (-1)^{(m-1)/2} + \tilde{C}_m = \]

\[ c_m 2^{-2(m-1)} \left[ (-1)^{(m-1)/2} \sum_{s=0}^{m-1} \left( \frac{m-1}{s} \right) \frac{(-1)^{m-1-s}}{2s -(m-1)} \sin((4s-2(m-1))\theta) + \right. \]

\[ \left( \frac{m-1}{(m-1)/2} \right) \theta + \tilde{C}_m, \]

where \( \tilde{C}_m, m = 1, 2, \ldots \) are real integration constants. As the CDF \( F_m(\theta) \) has its support on \((0, \frac{\pi}{2})\), we have \( F_m(0) = 0 \). In case \( m \) is odd, \( \tilde{C}_m = 0 \).

We implies

\[ c_m = \frac{2^{2m-1}}{(m-1)/2} \left( \frac{\pi}{2} \right). \]

• In case \( m \) is even, we can perform similar calculations, as follows:

\[ p_m(\theta) \] has the following primitive function

\[ c_m 2^{-(m-1)} \ Re \left[ (-i)^{m-1} \sum_{s=0}^{m-1} \left( \frac{m-1}{s} \right) \frac{(-1)^{m-1-s}}{i(4s-2(m-1))} e^{i(4s-2(m-1))\theta} + \tilde{C}_m \right] = \]
\[ c_m 2^{(-2m+1)} \Re[(-i)^m \sum_{s=0}^{m-1} \binom{m-1}{s} \frac{(-1)^{m-1-s}}{(2s-(m-1))} e^{i(4s-2(m-1))} + \tilde{C}_m] = \]

\[ c_m 2^{(-2m+1)} (-1)^{(m/2)} \sum_{s=0}^{m-1} \binom{m-1}{s} \frac{(-1)^{m-1-s}}{(2s-(m-1))} \cos((4s-2(m-1))\theta) + \tilde{C}_m. \]

\[ \tilde{C}_m = c_m 2^{(-2m+1)} (-1)^{(m/2)+1} \sum_{s=0}^{m-1} \binom{m-1}{s} \frac{1}{(2s-(m-1))} (-1)^{m-1-s}. \]

### 2.6 The Particular Compound Density

**Theorem 2.** Let \( m \) be odd and let the mixing density be given by \( p(\tau) = c_m \frac{\tau^{m-1}}{(1+\tau^2)^m} \), where \( c_m \) is the normalizing constant (as before). The compound density takes the form \( q(x) = c_m \frac{A_m(x)}{(x^2+N^2)^{N+m}} \), where \( A_m(x) \) is a polynomial with rational coefficients and degree at most \( 2(N+m-1) \).

**Proof.** We consider two following cases.

- **N even:**
  
  \[ q(x) = (C_0 + C_N)(x) + C_{N/2}(x) + \sum_{k=1}^{N-1} (C_k + C_{N-k})(x). \]

- **N odd:**
  
  \[ q(x) = (C_0 + C_N)(x) + \sum_{k=1}^{N-1} (C_k + C_{N-k})(x). \]

We have

\[ C_0 = \frac{1}{(1+\tau_0^2)^N \rho_m(\tau_0) \frac{1}{N} \mathbb{1}_{x \leq 0}} = c_m \frac{\tau_0^{m-1}}{(1+\tau_0^2)^{N+m}} \frac{1}{N} \mathbb{1}_{x \leq 0} = \]

\[ c_m (-1)^{m-1} \frac{x^{m-1} N(2N+m)}{(x^2+N^2)^{N+m}} \mathbb{1}_{x \leq 0} = c_m \frac{x^{m-1} N(2N+m)}{(x^2+N^2)^{N+m}} \mathbb{1}_{x \leq 0}. \]

Here we have used \((-1)^{m-1} = 1\) as \( m \) is odd.

\[ C_N = \frac{(\tau_N^2)^N}{(1+\tau_N^2)^N \rho_m(\tau_N) \frac{N}{x^2} \mathbb{1}_{x > 0}} = c_m \frac{\tau_N^{2N+m-1}}{(1+\tau_N^2)^{N+m}} \frac{N}{x^2} \mathbb{1}_{x > 0} = \]

\[ c_m \frac{x^{m-1} N(2N+m)}{(x^2+N^2)^{N+m}} \mathbb{1}_{x > 0}. \]
It follows that the first term \((C_0 + C_N)(x)\) can be written explicitly as follows for all real values of \(x\):

\[
(C_0 + C_N)(x) = c_m \frac{N^{2N+m}x^{m-1}}{(x^2 + N^2)^{N+m}} = c_m \frac{A_m(x, 0)}{(x^2 + N^2)^{N+m}},
\]

where \(A_m(x, 0) := N^{2N+m}x^{m-1}\). Recall that \((m-1)\) is even, so \((C_0 + C_N)(x)\) is even w.r.t \(x\).

Now we consider the case \(1 \leq k \leq N\). Observe that

\[
\left(\frac{-x + y_k}{2(N - k)}\right)(\frac{x + y_k}{2k}) = \frac{-x^2 + y_k^2}{4(N - k)k} = \frac{-x^2 + x^2 + 4k(N - k)}{4k(N - k)} = 1.
\]

Hence, as \(\tau_k = \frac{-x + y_k}{2(N - k)}\) then \(\tau_k^{-1} = \frac{x + y_k}{2k}\) and \(\tau_N^{-1} = \frac{x + y_k}{2(N - k)}\). We rewrite \(C_{N-k}\) in terms of \(\tau_N^{-1}\) as follows:

\[
C_{N-k} = \binom{N}{N - k} \left(\frac{\tau_N^{-1}}{1 + \tau_N^{-1}}\right)^{N+k} = c_m \frac{N^m \tau_N^{-1}}{(1 + \tau_N^{-1})^N} \binom{N}{N - k} \frac{\tau_N^{-1}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{N - k} \left(\frac{\tau_N^{-1}^{N-k}}{1 + \tau_N^{-1}^{N-k}}\right)^{N} \tau_{N-k}^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{N - k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{N - k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{N - k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

Notice that \(y_{N-k} = y_k\) and \(\binom{N}{N - k} = \binom{N}{k}\). The term \(C_k + C_{N-k}\) can be defined w.r.t \(x\) and \(y_k\) as follows:

\[
(C_k + C_{N-k})(x, y_k) = c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]

\[
= c_m \frac{\binom{N}{k} \left(\frac{\tau_N^{-1}^{N-k} + \frac{m}{2} \tau_{N-k}^{-1}}{1 + \tau_N^{-1}^{N-k}}\right)^{N^m \tau_N^{-1}}}{(1 + \tau_N^{-1}^{N-k})^N} \binom{N}{k} \frac{\tau_N^{-1}^{N^m \tau_N^{-1}}}{y_{N-k}}
\]
where \( N(x, y_k) = (4(N-k)^2 + (-x+y_k)^2)^{N+m}(x+y_k)^{2k+m} + (4(N-k)^2 + (x+y_k)^2)^{N+m}(-x+y_k)^{2k+m} \).

\[
D(x, y_k) = \left( (4(N-k)^2 + x^2 + y_k^2)^2 - 4x^2 y_k^2 \right)^{N+m} \cdot y_k.
\]

Observe that \( N(x, y_k) = -N(x, -y_k) \) and \( D(x, y_k) = -D(x, -y_k) \), so \( \frac{N(x, y_k)}{D(x, y_k)} = \frac{N(x, -y_k)}{D(x, -y_k)} \).

Hence, \((C_k + C_{N-k})(x, y_k)\) is even in \( y_k \). We can see that

\[
N(x, 0) = (4(N-k)^2 + (x)^2)^{N+m}(x)^{2k+m} - (4(N-k)^2 + (x)^2)^{N+m}(x)^{2k+m} = 0
\]

Hence, we infer that \( N(x, y_k) \) is divisible by \( y_k \), so both numerator and denominator are divisible by \( y_k \). We have \( D(x, y_k)/y_k \) and \( C_k + C_{N-k} \) both are even in \( y_k \), so \( N(x, y_k)/y_k \) is even in \( y_k \). This implies that we can express the new numerator \( N(x, y_k)/y_k \) and the new denominator \( D(x, y_k)/y_k \) both as polynomial in terms of powers of \( x \) and powers of \( y_k^2 \).

Replacing \( y_k^2 = x^2 + 4k(N-k) \), we can conclude that \((C_k + C_{N-k})(x)\) is rational in \( x \). In a similar way we can see that \((C_k + C_{N-k})(x)\) is an even function of \( x \).

Observe that

\[
\left( (4(N-k)^2 + x^2 + y_k^2)^2 - 4x^2 y_k^2 \right) = \\
(4(N-k)^2 + x^2 + x^2 + 4k(N-k)^2)^2 - 4x^2 (x^2 + 4k(N-k)) = \\
4\{(2(N-k)^2 + x^2 + 2k(N-k))^2 - x^4 - 4k(N-k)x^2\} = \\
4\{(2N(N-k) + x^2)^2 - x^4 - 4k(N-k)x^2\} = \\
4\{x^4 + 4N(N-k)x^2 + 4N^2(N-k)^2 - x^4 - 4k(N-k)x^2\} = \\
16(N-k)\{(N-k)x^2 + N^2(N-k)\} = 16(N-k)^2\{x^2 + N^2\}.
\]

Hence,

\[
(C_k + C_{N-k})(x, y_k) = c_m \frac{\tilde{N}(x, y_k)}{\binom{2N+2k+3m(N-k)^2k+m}{N} \cdot (x^2 + N^2)^{N+m}},
\]

where

\[
\tilde{N}(x, y_k) = \frac{N(x, y_k)}{y_k}, \quad d_k := \frac{2^{2N+2k+3m(N-k)^2k+m}}{\binom{N}{k}}, \quad A_m(x, k) := \frac{\tilde{N}(x, y_k)}{d_k}.
\]
Therefore, the term \((C_k + C_{N-k})(x)\) can hence be written as
\[
(C_k + C_{N-k})(x) = c_m \frac{A_m(x, k)}{(x^2 + N^2)^{N+m}},
\]
where \(A_m(x, k)\) is a polynomial with rational coefficients.

Note that in case \(k = N/2\) this formula implies that \(C_k(x) = \frac{1}{2}c_m \frac{A_m(x, k)}{(x^2 + N^2)^{N+m}}\). The compound density \(q(x)\) is an even rational function with common denominator \((x^2 + N^2)^{N+m}\) because it is the sum of even rational functions with the same denominator. The numerator of \(q/c_m\) is the sum of polynomials with rational coefficients, hence is a polynomial with rational coefficients. Notice that as \(q\) has integral one over the real line and each of the \((C_k + C_{N-k})(x)\) functions is non-negative, each such function is integrable and hence its codegree must be at least 2. The same argument holds for \(q\) itself and so the numerator degree of \(q\) will be less than or equal to \(2(N + m - 1)\).

To compute \(A_m(x, k)\), one could use algebraic manipulation with Euler substitution to eliminate all the occurrences of square roots. Alternatively, one could compute \(A_m(x, k)\) using a Lagrange interpolation technique that we will now explain. We need to take \((N + m)\) interpolation points \(x_k^i, i = 1, 2, \ldots, N + m\) to approximate \(A_m(x, k)\) because the degree of numerator \(A_m(x, k)\) is at most \(2(N+m-1)\) and \(A_m(x, k)\) is even. Here \(k\) is fixed for each term \((C_k + C_{N-k})(x)\). We can calculate the values of \(A_m(x_k^i, k)\) by noting that
\[
A_m(x_k^i, k) = \frac{1}{c_m} \frac{(C_k + C_{N-k})(x_k^i)(x_k^i)^2 + N^2)^{N+m}}{N+m}.
\]

By applying Lagrange interpolation method, we can obtain
\[
\begin{bmatrix}
  A_m(x_1^k, k) \\
  A_m(x_2^k, k) \\
  \vdots \\
  A_m(x_{N+m}^k, k)
\end{bmatrix}
= \begin{bmatrix}
  1 & (x_1^k)^2 & (x_1^k)^4 & \cdots & (x_1^k)^{2(N+m-1)} \\
  1 & (x_2^k)^2 & (x_2^k)^4 & \cdots & (x_2^k)^{2(N+m-1)} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  1 & (x_{N+m}^k)^2 & (x_{N+m}^k)^4 & \cdots & (x_{N+m}^k)^{2(N+m-1)}
\end{bmatrix}
\begin{bmatrix}
  a_0 \\
  a_2 \\
  \vdots \\
  a_{2(N+m-1)}
\end{bmatrix}
\]

The matrix is a \((N+m) \times (N+m)\) Vandermonde matrix of interpolation points. It is known to be non-singular as the interpolation points will be distinct. The Lagrange matrix is the known inverse matrix of this Vandermonde matrix, so we can obtain the solution by using the Lagrange coefficients explicitly. Alternatively we can solve this linear system of
equations directly by standard methods.

Recall $A_m(x, k) = \tilde{N}(x, y_k)/d_k$, where $\tilde{N}(x, y) = \frac{N(x, y)}{y}$ is a known two-variable polynomial in $x$ and $y$ with integer coefficients and $d_k$ is a known integer. To be able to get the rational coefficients of $A_m(x, k)$ exactly, we need to take the interpolation points such that both $x^i_k$ and $y_k(x^i_k)$ are rational! That is indeed possible as we will now show. Our approach will be based on an Euler substitution (known from the theory of integration).

Let $z_k := -x + y_k = -x + \sqrt{x^2 + 4k(N - k)}$ then $z_k + x = y_k$. It implies that

$$z_k^2 + x^2 + 2z_kx = x^2 + 4k(N - k) \iff 2z_kx = 4k(N - k) - z_k^2 \iff x = \frac{2k(N - k)}{z_k} - \frac{1}{2}z_k.$$

It follows that $y_k = x + z_k = \frac{2k(N-k)}{z_k} + \frac{1}{2}z_k$.

Note that if we choose $z_k$ rational and non-zero ($z_k \in \mathbb{Q} \setminus \{0\}$) then both $x$ and $y_k$ will be rational. Furthermore, for any $\hat{x} \in \mathbb{R}$ one can calculate $\hat{z}_k = -\hat{x} + \sqrt{\hat{x}^2 + 4k(N - k)} > 0$ and take a positive rational number $z_k$ arbitrarily close to $\hat{z}_k$ and compute the corresponding rational values of $x$ and $y_k$. By taking $z_k$ sufficiently close to $\hat{z}_k$, the corresponding values $x$ and $y_k$ will be as close as is desired to $\hat{x}$ and $\hat{y}_k$. (Warning: care must be taken for cases in which $\hat{z}_k$ is close to zero).

The Lagrangian interpolation method now requires us to solve a Vandermonde-type linear system of $(N+m)$ equations with only rational coefficients. The solution will be a vector of rational numbers in $\mathbb{Q}^{N+m}$. How to solve such systems in case $N+m$ is large is an active area of research in which considerable advances have been made [31]. In our project, we have relied on existing routines in symbolic computation.

**Remark 1.**

A random variable $X$ has the mean at zero and the variance at $N$ with the PDF $q(x)$. Then $Z := \frac{X}{\sqrt{N}}$ has the mean at zero and the variance at 1 with the PDF $\sqrt{N}q(\frac{z}{\sqrt{N}})$.

**Remark 2.**

MC-Tree allows to generate probability parameters and other parameters randomly with
a well-known chosen probability distribution. MC-Tree holds all benefits of both the MC method and the tree method. Also, one advantage of the MC-Tree approach is that it allows us to work with the confidence interval of the MC simulations. If we use only the tree method, we can not have the confidence interval and its benefit. The confidence interval depends on the number of simulations.

**Remark 3.**

For numerical implementation in section 5, we only search among odd values of m as these are the only ones giving a rational compound density. Based on our numerical results on entropy, but also on the Kullback-Leibler (KL) Divergence, as well as $L^1$ distance, the best choice for the integer $m$ in our class of mixing densities is at $m = 9$.

## 3 The Correction Techniques

### 3.1 Usage of a Bias-Correction

To compute a European option, a price process $S_t$ is modeled by the geometric Brownian motion under its associated "risk-neutral" measure $Q$

$$dS_t = rS_t dt + \sigma S_t dW_t,$$

where $r \in \mathbb{R}$ is the (constant) interest rate, $\sigma \in \mathbb{R}$ represents the diffusion coefficient, and $W$ is a standard Brownian motion process under $Q$. Using Itô’s lemma with $f(S) = \log(S)$ gives a classic result, in which the process $\log(S)$ follows the normal distribution $\mathcal{N}((r - \sigma^2/2)T, \sigma^2 T)$ on any interval of length $T$. This process can be approximated by a binomial tree.

We consider for instance the payoff functions $\pi(X) = \text{Max}(e^X - K, 0)$ for a call option or $\pi(X) = \text{Max}(K - e^X, 0)$ for a put option, where $X = \log(S)$.

For option pricing one typically uses the multiplicative tree (as mentioned before). Using this a multiplicative upward move, $u_1$, and a multiplicative downward move, $d_1$, are defined
\[ u_1 = \exp\{u\sigma\sqrt{\delta t} + (r - \sigma^2/2)\delta t\}, \quad d_1 = \exp\{d\sigma\sqrt{\delta t} + (r - \sigma^2/2)\delta t\}. \]

Here \( \delta t > 0 \) denotes the time-step length and translation and scaling have been applied to introduce the volatility parameter \( \sigma \) and the drift term \( r \). Using bias-corrected directions (see [30]) gives

\[ \tilde{u}_1 := u_1 e^{-\lambda \delta t}, \quad \tilde{d}_1 := d_1 e^{-\lambda \delta t}, \]

where the real number \( \lambda \) is solved from \( p\tilde{u}_1 + (1 - p)\tilde{d}_1 = e^{r\delta t} \). We obtain

\[ \lambda = \frac{\log(pu_1 + (1 - p)d_1)}{\delta t} - r. \]

This correction amounts to replacing \( \tilde{\mu} = r - \frac{1}{2}\sigma^2 \) by \( \hat{\mu} = r - \frac{1}{2}\sigma^2 + \lambda \). The resulting tree model is complete and free of arbitrage and has the probability given by \( p_1 \) and \( p_2 \) as the risk-neutral probability in the tree (see [30]).

### 3.2 Usage of a Distribution Correction Factor

Consider the compound density now with the appropriate variance and mean (obtained by scaling and translation as in the previous section). With some abuse of notation we will denote this again by \( q \). Suppose we use a mixing density, say \( p_m \) with \( m = 9 \) for instance. Then the compound density \( q \) will be close to Gaussian, in the sense that it has high entropy, especially for deeper trees, but it is not exactly equal to the corresponding Gaussian density with same mean and variance. To compensate for that, one can employ a distribution correction factor. This technique is known from the Monte Carlo method of importance sampling. The distribution correction factor, which we will denote by \( C(x) \), can be derived, in the context of option pricing, as follows:

Let \( P \) denotes the price of an European option with payoff \( \pi(X) \) at time \( T \), then we have

\[
\begin{align*}
P &= e^{-rT}\mathbb{E}_Q[\pi(X)] = e^{-rT}\int \pi(x)f(x)dx = e^{-rT}\int [\pi(x)\frac{f(x)}{q(x)}]q(x)dx \\
&= e^{-rT}\int [\pi(x)C(x)]q(x)dx = e^{-rT}\mathbb{E}_q[\pi(X)C(X)],
\end{align*}
\]
where q(x) is the compound density and we have the Gaussian density \( f(x) \sim N((r - \frac{1}{2}\sigma^2)T, \sigma^2T) \). In this way we get an exact Monte Carlo method for the pricing of European options that depend only on the asset price at expiry, in the sense that the compensated compound density is exactly Gaussian.

4 Unilateral CVA

Modelling CVA is complicated because it consists of at least three components: Probability of Default (PD), Loss Given Default (LGD), and Exposure at Default (EAD). We can make some standard assumptions. Assume that LGD and discount factors are nonrandom. We assume the possible default event of the counterparty and the value of a netting set are uncorrelated, which is an assumption of unilateral CVA. We will use an intensity default model \([9]\) to calculate the default probability of a counterparty. The exposure calculations depend highly on the complexity of the derivatives in the portfolio. Modelling CVA of an American option is a challenge due to the complexity of CVA calculations and the characteristics of the American option. We present an algorithm for the calculation of the expected exposure in the formula of the unilateral CVA for the American put option using MC-Tree in the next section. Our method of calculations for the CVA of an American option is not known in the literature, to the best knowledge of the authors.

4.1 Tree Approach for CVA

We develop the algorithm to calculate CVA on an American put option, using the MC-tree method, as follows:

**Step 1:** Run the tree backward to compute the American option value at each node. Label each node either C for continuation or N for no continuation. Let

\[
\mathbb{1}_C = \begin{cases} 
1 & \text{if node label = "C"}, \\
0 & \text{if node label = "N"}.
\end{cases}
\]
Step 2: Run the tree forward to compute the probabilities of the American option reaching a given node.

\[ P(A) = p \times P(\text{Lower Predecessor}) + (1 - p) \times P(\text{Upper Predecessor}) \]

Step 3: Run the tree backward to compute the expected exposure at time step in the tree using the probabilities in step 2.

4.2 MC-Tree Approach for CVA

The tree approach for CVA will produce a CVA value for any given tree. The CVA value from the MC-Tree approach is the mean of all CVA values of all trees.

5 Numerical Results

5.1 Pricing European Options

We will present the numerical results of some experiments to pricing the European option and compare MC-Tree with the usage of the bias-correction and the distribution correction factor with the Monte Carlo (MC) method and popular binomial tree models. We can obtain the analytical solution from the well-known Black-Scholes model. The error is the difference between the model value and the analytical solution.

All experiments were conducted on the machine i5-10210U, 8GB memory, R i386 3.5.1. The following parameters are used through all numerical experiments.
• Strike price $K = 95$.

• Expiration time $T = 1$.

• Risk-free rate $r = 0.03$.

• Volatility $\sigma = 0.2$.

It is verified that the put-call parity is hold for the MC-Tree with the bias-correction, as shown in Figure 1.

![Plot of Put-Call parity](image)

Figure 1: Verification of Put-Call Parity

5.1.1 Comparison to MC Method

Table 1 and table 2 show that MC-Tree is more accurate than MC method. The usage of the distribution correction factor in option pricing improves the accuracy dramatically, resulting in the exact analytical solution at the tree depth $N=50$ or $100$, and the number of simulations $M=100000$.

Table 3 shows the results from both methods for similar computation time. It is evident
| S  | N  | Method | Mean   | SD    | CI              | AS       |
|----|----|--------|--------|-------|-----------------|----------|
| 100| 50 | Corr   | 12.1798| 0.025 | (12.17965, 12.17995) | 12.1797 |
| 50 |    | Bias   | 12.1905| 0.0279| (12.1903, 12.1907)   |          |
| 100|    | Corr   | 12.1797| 0.0123| (12.17962, 12.17978) |          |
| 100|    | Bias   | 12.1851| 0.0155| (12.1850, 12.1852)   |          |
|    |    | MC     | 12.1867| 15.6215|(12.0899, 12.2835)    |          |
| 90 | 50 | Corr   | 6.2125 | 0.071 | (6.2121, 6.2130)     | 6.2125   |
| 50 |    | Bias   | 6.2230 | 0.0596| (6.2226, 6.2233)     |          |
| 100|    | Corr   | 6.2125 | 0.0463| (6.212213, 6.212787) |          |
| 100|    | Bias   | 6.2177 | 0.0401| (6.2175, 6.2180)     |          |
|    |    | MC     | 6.2143 | 11.1305|(6.1453, 6.2833)      |          |

Table 1: Accuracy Comparison between MC-Tree and MC Method in Pricing European Call Option. Bias: MC-Tree with the usage of Bias-Correction; Corr: MC-Tree with the usage of the distribution correction factor.

| S  | N  | Method | Mean   | SD    | CI              | AS       |
|----|----|--------|--------|-------|-----------------|----------|
| 100| 50 | Corr   | 4.3720 | 0.0324| (4.3718, 4.3722) | 4.3720   |
| 50 |    | Bias   | 4.3828 | 0.0279| (4.3827, 4.3830) |          |
| 100|    | Corr   | 4.3720 | 0.0185| (4.3719, 4.3721) |          |
| 100|    | Bias   | 4.3774 | 0.0155| (4.3773, 4.3775) |          |
|    |    | MC     | 4.4107 | 7.6584|(4.3633, 4.4582)  |          |
| 90 | 50 | Corr   | 8.4048 | 0.0503| (8.4045, 8.4051) | 8.4048   |
| 50 |    | Bias   | 8.4153 | 0.0596| (8.4149, 8.4157) |          |
| 100|    | Corr   | 8.4048 | 0.0345| (8.4046, 8.4050) |          |
| 100|    | Bias   | 8.4101 | 0.0401| (8.4098, 8.4103) |          |
|    |    | MC     | 8.4352 | 10.1748|(8.3721, 8.4982)  |          |

Table 2: Accuracy Comparison between MC-Tree and MC Method in Pricing European Put Option. Bias: MC-Tree with the usage of Bias-Correction; Corr: MC-Tree with the usage of the distribution correction factor.

from table 3 that the MC-Tree model is still more accurate than the MC method, even with the similar computation time.

5.1.2 Comparison to Binomial Models

Table 4 and table 5 show that MC-Tree with the usage of the distribution correction factor performs best. Option price from MC-Tree quicker converges to the analytical price when increasing the tree depth. It is evident from figure 2 that CRR and JR model is less stable and more volatile than the MC-Tree model as the tree depth increases.
|                  | MC-Tree with correction | MC       |
|------------------|-------------------------|----------|
| Option Price     | 12.1798                 | 12.1800  |
| SD               | 0.025                   | 15.6149  |
| CI               | (12.17965, 12.17995)    | (12.17117, 12.18883) |
| Error            | 5.7e-05                 | 3e-04    |
| Computation Time (Seconds) | 34.92103           | 35.50149 |
| M                | 100000                  | 12000000 |

Table 3: Error and Computation time of European call Option with similar running time at N=50.

| S    | Method | N = 50 |       |       |       |       |       |
|------|--------|--------|-------|-------|-------|-------|-------|
|      |        | Mean   | SD    | CI    | Mean  | SD    | CI    |
| 100  | Corr   | 12.1798| 0.025 | (12.1797, 12.1800) | 12.1797 | 0.0123 | (12.1796, 12.1798) |
|      | Bias   | 12.1905| 0.0279| (12.1903, 12.1907) | 12.1851 | 0.0155 | (12.1850, 12.1852) |
|      | CRR    | 12.1733| 12.1677|       |       |       |       |
|      | JR     | 12.1733| 12.1677|       |       |       |       |
| 90   | Corr   | 6.2125 | 0.071 | (6.2121, 6.2130) | 6.2125 | 0.0463 | (6.2122, 6.2128) |
|      | Bias   | 6.2230 | 0.0596| (6.2226, 6.2233) | 6.2177 | 0.0401 | (6.2175, 6.2180) |
|      | CRR    | 6.1912 | 6.2281|       |       |       |       |
|      | JR     | 6.1912 | 6.2281|       |       |       |       |

Table 4: Accuracy Comparison between MC-Tree and Binomial models in Pricing European Call Option. Bias: MC-Tree with the usage of Bias-Correction; Corr: MC-Tree with the usage of the distribution correction factor.

Then, we can obtain the mean squared error (MSE) for different models from the range of tree depth from 1 to 100 in Table 6. MSE from the MC-Tree is the lowest among models.

Figure 3 plots the European call prices with a range of M and the tree depth from 20 to 200. Clearly, the price is more stable with a rather large M.
| S   | Method | $N = 50$ |            | $N = 100$ | AS  |
|-----|--------|----------|------------|------------|-----|
|     |        | Mean     | SD         | CI         | Mean | SD   | CI   |
| 100 | Corr   | 4.3720   | 0.0324     | (4.3718,   | 4.3720 | 0.0185 | (4.3719,   |
|     |        |          |            | 4.3722)    |       |       | 4.3721) |
|     | Bias   | 4.3828   | 0.0279     | (4.3827,   | 4.3774 | 0.0155 | (4.3773,   |
|     |        |          |            | 4.3830)    |       |       | 4.3775) |
|     | CRR    | 4.3657   |            | 4.3846     |       |       | 4.3830 |
|     | JR     | 4.3600   |            | 4.3907     |       |       | 4.3846 |
| 90  | Corr   | 8.4048   | 0.0503     | (8.4045,   | 8.4048 | 0.0345 | (8.4046,   |
|     |        |          |            | 8.4051)    |       |       | 8.4050) |
|     | Bias   | 8.4153   | 0.0596     | (8.4149,   | 8.4101 | 0.0401 | (8.4098,   |
|     |        |          |            | 8.4157)    |       |       | 8.4103) |
|     | CRR    | 8.3835   |            | 8.4206     |       |       | 8.4206 |
|     | JR     | 8.4204   |            | 8.4008     |       |       | 8.4008 |

Table 5: Accuracy Comparison between MC-Tree and Binomial models in Pricing European Put Option. Bias: MC-Tree with the usage of Bias-Correction; Corr: MC-Tree with the usage of the distribution correction factor.

| Models | MC-Tree | CRR      | JR      |
|--------|---------|----------|---------|
| MSE    | 0.00015354 | 0.001074532 | 0.001015309 |

Table 6: Mean-Squared Error for Various Models

5.2 Pricing American Put Option

We will present the numerical results of some experiments to the American Put option and compare them with the LSM and popular binomial tree models to gain some insight into the performance of the MC-Tree Method. The quadratic polynomial is used in the regression model [23].

All numerical experiments use the same parameters as mentioned in the previous section: pricing European option, except $M=2000$. We consider various examples of American option valuation and compare our method with the LSM, CRR, and JR.

5.2.1 Comparison to LSM Method

We will compare the standard deviation of the two methods to understand their accuracy. Table [7] shows the mean and the standard deviation from simulation results with the initial
stock price at 100, and the "true" price at 4.5415. The "true" price of an American put option is obtained by the convergent binomial method with the depth of tree at 50,000.

As shown in the table [7], the standard deviation of the LSM is much larger than the

|                | MC-Tree | LSM |
|----------------|---------|-----|
| Mean           | 4.5483  | 4.5782 |
| Standard Deviation | 0.0319  | 7.1828 |

Table 7: Mean and Standard Deviation of American Put Option.
one of the MC-Tree method when their means are similar. The LSM needs a significant increase in the number of replications to improve accuracy, which leads to an increase in the computation time per simulation.

It is evident from table 8 that it is not sufficiently good for the LSM to obtain a small standard deviation as the MC-Tree method, even that the computation time of both are almost the same. It is concluded that the MC-Tree method provides us with more accuracy than the LSM at a similar computational cost.

|                      | MC-Tree | LSM     |
|----------------------|---------|---------|
| Mean                 | 4.5483  | 4.5274  |
| Standard Deviation   | 0.0319  | 0.8465  |
| Error                | 0.0068  | 0.0141  |
| Computation Time (Seconds) | 10.4658 | 10.5423 |
| M                    | 2000    | 120000  |

Table 8: Mean and standard deviation of American put Option with similar running time.

5.2.2 Comparison to Binomial Models

We will use the same model parameters, as mentioned in the previous section. As shown in table 9 and table 10, the option prices among models are insignificantly different, and
the MC-Tree method produces the smallest error. It can be concluded that the MC-Tree method performs better than other methods: CRR, JR in terms of accuracy using the same tree-depth. Amin and Khanna [1] proved that American option prices of the discrete model also converge to the corresponding value of the continuous-time model under fairly general conditions. It means that the "true" price can be obtained by increasing the tree depth to infinity. Therefore, we also compare results with the "true" prices. Figure 4 shows two

| Stock Price | MC-Tree | CRR  | JR   | "True" Price |
|-------------|---------|------|------|--------------|
| 95          | 6.4140  | 6.3966 | 6.4141 | 6.4058       |
| 97          | 5.6058  | 5.6148 | 5.6080 | 5.5973       |
| 100         | 4.5484  | 4.5511 | 4.5583 | 4.5415       |
| 102         | 3.9409  | 3.9433 | 3.9240 | 3.9338       |
| 104         | 3.4007  | 3.4034 | 3.4111 | 3.3960       |

Table 9: Option Prices

| Stock Price | MC-Tree | CRR  | JR   |
|-------------|---------|------|------|
| 95          | 0.0081  | 0.0093 | 0.0083 |
| 97          | 0.0084  | 0.0175 | 0.0107 |
| 100         | 0.0080  | 0.0096 | 0.0168 |
| 102         | 0.0070  | 0.0095 | 0.0098 |
| 104         | 0.0047  | 0.0073 | 0.0150 |

Table 10: Accuracy Comparison among Models

plots for prices of an American put option by the MC-Tree, CRR, and JR model when increasing the tree depth. The CRR and JR model both are more volatile than the MC-Tree model. The MC-Tree model is more stable and substantially less deviance from the overall downward, convergent toward the "true" price as N increases, compared with the CRR and JR model.
5.3 Numerical Results of CVA Calculations

The following parameters are used to estimate CVA of an American put option. Initial stock price $S_0=80$, strike price $K=100$, expiration time $T=1$, risk-free rate $r=0.03$, volatility $\sigma = 0.2$, dividend rate=0, recovery rate $R=0.4$, intensity of default $\lambda = 0.03$.

CVA value is convergent to 0.34 when we increase the number of simulation $M$, and the tree depth $N$, respectively.

6 Concluding Remarks

The method presents bounds on how close our results are to the "true" prices and shows the confidence interval containing the "true" one with a given (high) probability at 95%. Prices from MC-Tree method converge to the analytical solution or the "true" price. The
Table 11: CVA Values

| N   | M    | CVA   |
|-----|------|-------|
| 50  | 100  | 0.2447|
| 75  | 150  | 0.3013|
| 100 | 250  | 0.3104|
| 250 | 700  | 0.3282|
| 250 | 10000| 0.3392|
| 250 | 100000| 0.3440|
| 2000| 700  | 0.3414|
| 4000| 700  | 0.3414|

completeness of the model allows to provide hedging strategies.

We implemented numerical experiments of MC-Tree on pricing options and CVA Calculations. We can conclude that MC-Tree method is an efficient method to price options on single asset. The model can be applied to practical development in financial industry due to its high accuracy.

7 Further Research

In the future, we intend to present the MC-Tree method, which combines the MC method with the recombining multinomial tree based on Pascal simplex \([30]\) for pricing multi-assets options. The research will be the natural generalization of the MC-Tree method in this article.

A future research direction is to generalize the model to a real market with stochastic parameters. Another promising research direction is to develop fast hardware implementations of the MC-Tree, which could be useful for the financial industry, especially the derivative pricing and risk management industry. For example, one can use an FPGA architecture to do the tree calculations, see for instance \([24]\) where this is worked out for pricing multi-asset options. This could be combined with our MC approach to arrive at fast MC-Tree results. The difference with what is done in that paper would be in the precomputing phase, where
we precompute the input parameters to the FPGA. The usage of the distribution correction factor brings very high accuracy in European option pricing using MC-Tree method. It will be interesting to investigate further how to use the distribution correction factor in pricing the American option.

**Acknowledgments** We want to acknowledge Marta Ferrario and Luca Pettinari for their earlier contributions to the theory of the MC-Tree method in this paper.

**References**

[1] Kaushik Amin and Ajay Khanna. Convergence of american option values from discrete-to continuous-time financial models 1. *Mathematical Finance*, 4(4):289–304, 1994.

[2] Basel-Committee et al. Basel iii: International regulatory framework for banks, 2017. Bank for International Settlements. Retrieved February 22, 2018, from [https://www.bis.org/bcbs/basel3.htm](https://www.bis.org/bcbs/basel3.htm).

[3] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[4] Phelim P Boyle. Options: A monte carlo approach. *Journal of Financial Economics*, 4(3):323–338, 1977.

[5] Phelim P Boyle. A lattice framework for option pricing with two state variables. *Journal of Financial and Quantitative Analysis*, pages 1–12, 1988.

[6] Phelim P Boyle, Jeremy Evnine, and Stephen Gibbs. Numerical evaluation of multivariate contingent claims. *The Review of Financial Studies*, 2(2):241–250, 1989.

[7] Paolo Brandimarte. *Numerical methods in finance and economics: a MATLAB-based introduction*. John Wiley & Sons, 2013.
[8] Damiano Brigo, Agostino Capponi, and Andrea Pallavicini. Arbitrage-free bilateral counterparty risk valuation under collateralization and application to credit default swaps. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 24(1):125–146, 2014.

[9] Damiano Brigo and Fabio Mercurio. *Interest rate models-theory and practice: with smile, inflation and credit*. Springer, Berlin, 2006.

[10] Mark Broadie and Paul Glasserman. Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8-9):1323–1352, 1997.

[11] Mark Broadie, Paul Glasserman, et al. A stochastic mesh method for pricing high-dimensional american options. *Journal of Computational Finance*, 7:35–72, 2004.

[12] Giovanni Cesari, John Aquilina, Niels Charpillon, Zlatko Filipovic, Gordon Lee, and Ion Manda. *Modelling, pricing, and hedging counterparty credit exposure: A technical guide*. Springer Science & Business Media, London, 2009.

[13] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. An intertemporal general equilibrium model of asset prices. *Econometrica: Journal of The Econometric Society*, pages 363–384, 1985.

[14] John C Cox, Stephen A Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.

[15] Darrell Duffie, Jun Pan, and Kenneth Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68(6):1343–1376, 2000.

[16] Darrell Duffie and Kenneth J Singleton. Modeling term structures of defaultable bonds. *The Review Of Financial Studies*, 12(4):687–720, 1999.

[17] Dan Franzén and Otto Sjöholm. Credit valuation adjustment: In theory and practice, 2014.
[18] John Hull and Alan White. Cva and wrong-way risk. *Financial Analysts Journal*, 68(5):58–69, 2012.

[19] Robert A Jarrow and Andrew Rudd. *Option pricing*. Irwin, Homewood, IL, 1983.

[20] Dietmar PJ Leisen and Matthias Reimer. Binomial models for option valuation-examining and improving convergence. *Applied Mathematical Finance*, 3(4):319–346, 1996.

[21] John Lintner. Security prices, risk, and maximal gains from diversification. *The Journal of Finance*, 20(4):587–615, 1965.

[22] Qian Liu. Calculation of credit valuation adjustment based on least square monte carlo methods. *Mathematical Problems in Engineering*, 2015, 2015.

[23] Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation: a simple least-squares approach. *The Review Of Financial Studies*, 14(1):113–147, 2001.

[24] Aidan O Mahony, Gil Zeidan, Bernard Hanzon, and Emanuel Popovici. A parallel and pipelined implementation of a pascal-simplex based two asset option pricer on fpga using opencl. In *2020 IEEE Nordic Circuits and Systems Conference (NorCAS)*, pages 1–6. IEEE, 2020.

[25] Valentin Mena Morales, Pierre-Henri Horrein, Amer Baghdadi, Erik Hochapfel, and Sandrine Vaton. Energy-efficient fpga implementation for binomial option pricing using opencl. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.

[26] Jan Mossin. Equilibrium in a capital asset market. *Econometrica: Journal of The Econometric Society*, pages 768–783, 1966.

[27] F. E. N. G. Qian. *Advanced estimation of credit valuation adjustment*. PhD thesis, Technische Universiteit Delft, April 2017.
[28] William F Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.

[29] Y. Shen. *Credit value adjustment for multi-asset options*. PhD thesis, Technische Universiteit Delft, 2014.

[30] Dirk Sierag and Bernard Hanzon. Pricing derivatives on multiple assets: recombining multinomial trees based on pascal’s simplex. *Annals of Operations Research*, 266(1-2):101–127, 2018.

[31] Daniel E Steffy. Exact solutions to linear systems of equations using output sensitive lifting. *ACM Communications in Computer Algebra*, 44(3/4):160–182, 2011.

[32] Yisong Tian. A modified lattice approach to option pricing. *The Journal of Futures Markets (1986-1998)*, 13(5):563, 1993.

[33] JA Tilley. Valuing american options in a path simulation model. *Insurance Mathematics and Economics*, 2(16):169, 1995.

[34] Steven H Zhu and Michael Pykhtin. A guide to modeling counterparty credit risk. *GARP Risk Review, July/August*, 2007.