Soft Comput (2007) 11:1115–1122
DOI 10.1007/s00500-007-0154-2

FOCUS

Using computational methodology to price European options with actual payoff distributions

Chieh-Chung Sheng · Hsiao-Ya Chiu · An-Pin Chen

Published online: 23 March 2007
© Springer-Verlag 2007

Abstract Most option pricing methods use mathematical distributions to approximate underlying asset behavior. However, pure mathematical distribution approaches have difficulty approximating the real distribution. This study first introduces an innovative computational method for pricing European options based on the real payoff distribution of the underlying asset. This computational approach can also be applied to applications related to expected value that require real distributions rather than mathematical distributions. This study makes the following contributions: (a) solving the risk neutral issue related to price options with real payoff distributions; (b) proposing a simple method for adjusting standard deviation based on the need to apply short term volatility to real world applications; (c) demonstrating an option pricing algorithm that is easy to apply to cross field applications.

Keywords Option pricing · Actual payoff distribution · Expected value

1 Introduction

An option is a tradable contract that confers the right, but not the obligation, to buy (call) or sell (put) an underlying asset at an agreed-upon price during a certain period or on a specific date. The value of such a contract is termed the option price or option value. Thus, an option price is the expected return of the underlying asset's final settlement price larger (call) or lesser (put) than the desired value (the agreed-upon price). Because option value is the expected return of a usually unpredictable underlying asset, option pricing methodologies have been widely adopted by cross fields applications that need to obtain the target's expected value under uncertainties. For example, the real options analysis (ROA) approach was widely adopted for assessing information technology investments during the early 1990s (Clemons 1991; Dos Santos 1991). Thus, improvements in option pricing methodology can significantly benefit expected value related applications.

Option pricing methods have been widely researched since the development of the Black-Scholes model (BS model) in Black and Scholes (1973). Numerous studies have attempted to relax the restrictive assumptions of the BS model by using various methodologies to approximate the real payoff distribution on assets in a risk-neutral manner and thus obtain the fair option price. Although it seems natural to obtain the option price based on real asset payoff distribution, this idea has rarely been implemented because the real distribution never behaves risk-neutrally. This characteristic limits the adoption of option pricing methodology in certain non-mathematical distribution applications because real world behavior frequently disobeys mathematical distributions. Furthermore, the time value decreasing speed of an option accelerates considerably (non-linearly) as the maturity date approaches, yielding large pricing error, but high-frequency (time interval less than 1 min) pricing methodologies have received little attention. This non-linear variation characteristic also limits high frequency applications. For example, applications with time to maturity less than one day are not

C.-C. Sheng(✉) · H.-Y. Chiu
The Department of Management Information,
Yu Da College of Business, No.168, Hsueh-fu Rd,
Chaochiao Township, Miaoli County, 361 Taiwan, ROC
e-mail: sam-cc@umail.hinet.net

H.-Y. Chiu
e-mail: kates.chiu@msa.hinet.net

A.-P. Chen
Institute of Information Management, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, 300 Taiwan, ROC
e-mail: apc@iim.nctu.edu.tw

suitable for traditional option pricing methodologies because its expected value varies significantly as the settlement time approaches. If an option pricing model can remove the above limitations, it will be more applicable not only in finance but also in cross field applications.

Accordingly, this study proposes a computational model for pricing European options (whose exercise is only permitted on expiry) using the real return of the underlying asset, and verifies the high-frequency pricing performance based on empirical investigation. Experimental results indicate not only that the real distribution pricing method outperforms the BS model, but also that modern computational methods can be adopted to implement possibility distribution applications rather than using mathematical distributions to approximate the real distribution via closed form formulas. According to the test results, the proposed model contributes significantly to overcoming the limitations of traditional options pricing models when adopted by numerous cross field applications. For example, researchers must determine whether their target index exhibits geometric Brownian motion with lognormal returns when integrating the BS model (or most option pricing models) to calculate the desired expected values, as Benaroch did in his research on IT investment risks (Benaroch 2002). However, there is no need to justify the target's distribution when using the proposed computational model.

The rest of this paper is organized as follows. Section 2 briefly discusses the traditional option pricing methodologies. Section 3 then discusses observations of asset real payoff distribution and the feasibility of applying the real distribution map to price European options. The pricing methodology and algorithms are also presented in this section. Next, Sect. 4 conducts an empirical study to verify effectiveness of applying real payoff distribution to price European options. Finally, conclusions and future research directions are presented in Sect. 5.

2 Backgrounds on option pricing models

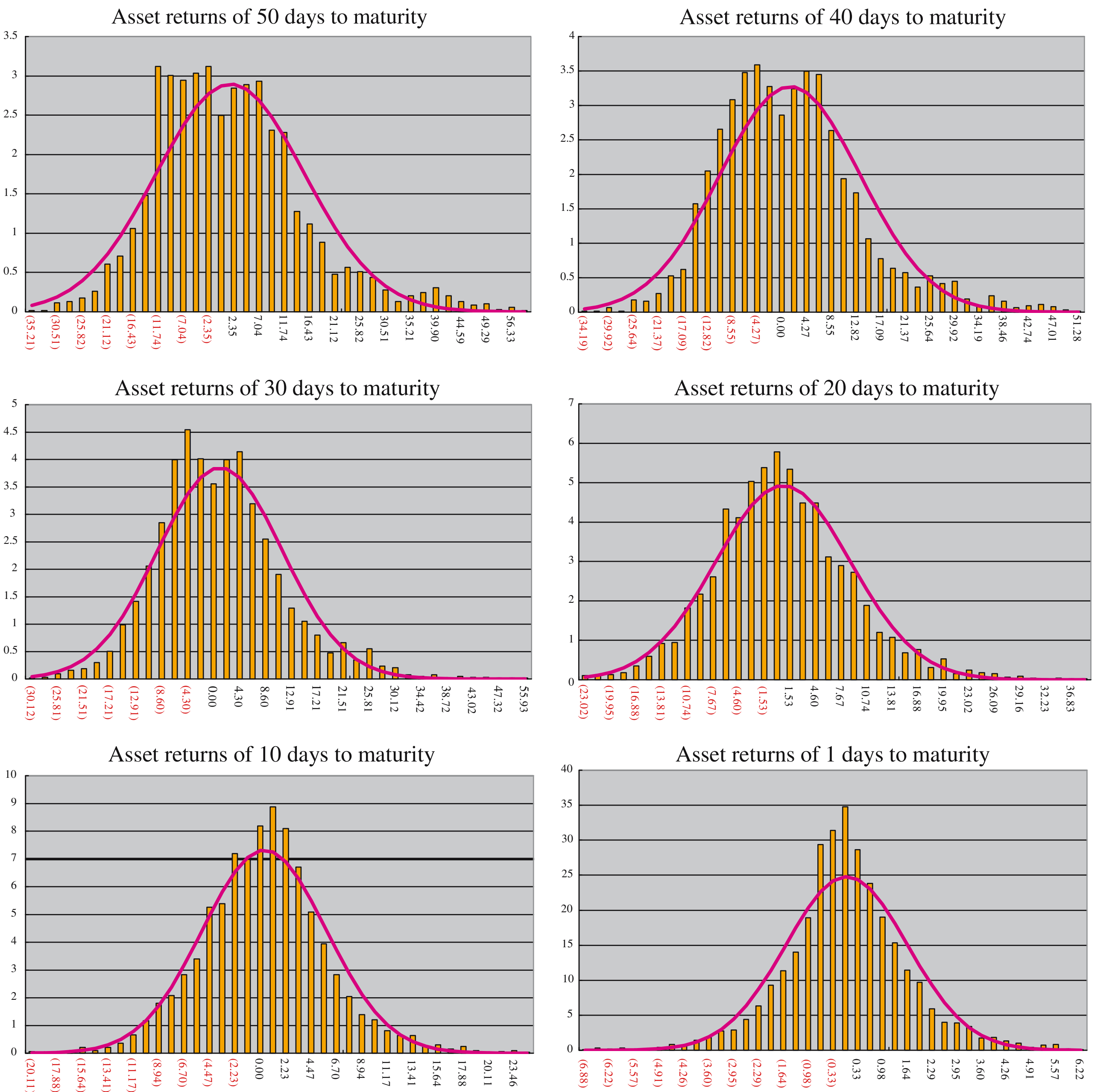
Cox and Ross (1976) established the option price as the expected payoff value discounted at the risk-free interest rate over the risk-neutral distribution of the underlying asset. However, applying the real payoff distribution rather than a mathematical risk-neutral distribution is difficult because the real distribution never behaves in a risk-neutral manner. Applying a distribution with non risk-neutral characteristic will violate put-call parity rules (Stoll 1969) because of the arbitrage possibilities associated with the derived put and call prices. A simple example is that if a distribution is risk-neutral then the mean value μ must equal zero. However, the μ in a real distribution rarely equals zero. The other difficulty in the application of real distribution is that it needs different distribution maps for different time to maturity. For example, at least n different distribution maps are needed to valuate the option price if it is n days before maturity. Thus, if the sampling data is huge then the pricing speed will be too slow for practical use. Additionally, short-term asset volatility is rarely consistent with that implied by the real distribution map, leading to significant pricing errors. Consequently, real asset return distribution cannot be practically used to obtain the option price, encouraging researchers to apply mathematically risk-neutral distributions instead. The most classical of these approaches is the BS model, which assumes that the payoff of the underlying asset follows the geometric Brownian motion and has a lognormal distribution with constant volatility and risk-free interest rate before maturity (Black and Scholes 1973). Since the development of the BS model, more realistic option pricing methodologies have been developed, including: (a) the stochastic interest-rate/volatility option model (Merton 1973; Amin and Jarrow 1992; Bates 1996); (b) jump-diffusion related models (Bates 1991; Madan et al. 1998); (c) Markovian models (Rubinstein 1994; Yacine and Andrew 1996); and (d) stochastic-volatility jump-diffusion models (Bates 1996; Scott 1997). However, all these models focus on identifying the "right" distributions and pricing options using close form formulas. Consequently, the mathematical distribution never perfectly fits any underlying asset's actual payoff distribution.

In computer science, attempts have also been made to price options using artificial intelligence models to improve options pricing performance. The most popular of these methods is the neural network approach. Unlike classical mathematical methodologies, a neural network is a non-parametric estimation technique which does not make any distributional assumptions regarding the underlying asset. Instead, this approach develops a model using sets of unknown parameters and lets the optimization routine seek the best fitting parameters to obtain the desired results. For example, Hutchinson et al. (1994) demonstrated that the neural network approach can be used to price S&P future options. Andrew Carverhill et al. (2003) followed this line of research and examined the best method of establishing and train a multi-layer perceptron neural network for option pricing and hedging. Meissner-Kawano (2001) also trained neural networks using option prices to address the smiling effect Meissner and Kawano (2001) associated with options' implied volatilities. All these works demonstrate that modern computational theories can offer alternative options pricing methods. However, few studies have used real payoff distribution to price options. Thus, this study focused on determining options price using the "real" payoff distribution obtained from a historical sample of the underlying asset.

Table 1 The real distribution maps compared to the normal distributions

1. The X-Axis is the nature log asset return rate in percentage and the Y-Axis is the possibility value in percentage.

2. The histograms represent the real payoff distribution and the curve lines represent the normal distribution.



3 Computational approach for pricing European options

This study proposes a computational method of pricing European options using high frequency time interval with one minute time ticks. High frequency examples are used to obtain large samples for verification purposes if the execution efficiency of this computation method can feasibly be applied to real world applications. The same concept can also be applied to price European options regardless of time interval.

3.1 Observations regarding real distributions

Most option pricing models use mathematical distributions. For example, the BS model assumes that underlying assets follow a geometric Brownian motion with lognormal returns. Meanwhile, other sophisticated option pricing methodologies like the stochastic volatility model apply a flexible distributional structure in which the correlation between volatility shocks and underlying stock returns controls the level of skewness, and use the volatility variation coefficient to control the kurtosis level (Bates 1996; Scott 1997). However, none of these mathematical distributions can describe underlying asset behavior in the real world.

To observe the real behaviour of the underlying assets, this study used sampling data for the period 03/01/2001–31/12/2003[1] from the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). Because most mathematical option pricing models discuss the underlying asset return distribution using lognormal related distributions (or with certain modifications), this study calculates the asset return rate as $\ln(\frac{P_t}{P_0})$ with different times to maturity where P_0 is the original price and P_t represents the price after t days. The actual distributions are compared with the normal distributions as listed in Table 1.

From Table 1, the real payoff distribution of the asset (TAIEX) varies with days-to-maturity. That is, the real distribution is time variant. The most interesting finding is that the actual distribution exhibits twin-peak phenomenon in 30, 40 and 50 days to maturity distribution maps. Restated, when days to maturity exceeds 30, the real asset return rate distribution displays two peaks. This twin peak phenomenon has received little attention from academics.

The real distribution clearly shows that mathematical distribution approaches have difficulty obtaining precise option price (at least for the Taiwan stock market), because the actual distribution varies according to time to maturity. The time variant distribution issue limits the use of fixed mathematical distribution pattern across the entire time to maturity range because variation in time to maturity requires the option pricing model to apply different distributions. However, it is difficult for mathematical models to apply different distributions for different time to maturity. Furthermore, behaviour may differ among assets and markets, so a mathematical model must apply different distributions to maximize its pricing performance for different assets or different markets. Another issue is that the actual payoff distribution, like the time variant twin peaks distributions is difficult to describe using mathematical distributions. This issue also limits the cross-field applications of using the traditional option pricing models.

[1] The tick transaction samples from 2004 to 2006 were lost due to a hard disk crash. The tick data was real-time collected by our financial lab server via a real-time data source, making data purchase or recovery difficult. Consequently, the 2001–2003 samples were used for the demonstration to achieve a consistent sample distribution.

3.2 The computational approach

Option price is the expected value of the payoff discounted at the risk-free interest rate over the risk-neutral distribution of the underlying asset. Thus, given the price S and an agreed-upon price K for the underlying asset applicable during a certain period T, the option value can be described as follows:

$$C = E(\max(S - K, 0))$$
$$P = E(\max(K - S, 0))$$

where C denotes the call option price, P represents the put option price, and $E(.)$ is the expected value.

In the real world the price of most assets varies continuously, and this variation is described as volatility σ. An option pricing model calculates C or P of the underlying asset under the circumstances (S, K, σ, T, r).

Assume I days of sampling data, with each day containing J time ticks. Then for each sample of ith day and jth time tick $X_{i,j}$, the tick payoff rate $R_{i,j}$ is

$$R_{i,j} = \begin{cases} \frac{X_{i+1,1}}{X_{i,j}}, & \text{if the final settlement price is determined by the opening price on the final settlement day} \\ \frac{X_{i+1,n}}{X_{i,j}}, & \text{if the final settlement price is determined by the closing price on the final settlement day} \end{cases}$$

Notably, $R_{i,j}$ can also be represented as $ln(X_{i+1,1}/X_{i,j})$ or $ln(X_{i+1,n}/X_{i,j})$ based on the assumptions of the BS model. However, the difference of applying logarithm or simple payoff rate is minor for high frequency applications. This study avoids unnecessary use of floating point functions to increase execution speed.

The payoff rate can be preprocessed and stored in a database table for further use in achieving a reasonable execution speed when calculating option prices for practical use.

Assume an option matures the next day and has strike price S, final settlement price S_t, exercise price K and current time-tick j. Given m sampling days (which can only generate $m-1$ sample entries), the call price C can be approximated as follows:

$$C(S, K, j) = E(\max(S_t - K, 0)) = \frac{\sum_{i=1}^{m-1} S \times \max(R_{i,j} - \frac{K}{S}, 0)}{m-1}$$

Similarly, the Put price P can be approximated as follows:

$$P(S, K, j) = E(\max(K - S_t, 0)) = \frac{\sum_{i=1}^{m-1} S \times \max(\frac{K}{S} - R_{i,j}, 0)}{m-1}$$

Consider the riskless interest rate r with time to maturity τ, the Call/Put price can be represented as:

$$C(S, K, j, r, \tau) = \frac{\sum_{i=1}^{m-1} S \times \max(R_{i,j} - \frac{e^{-r\tau}K}{S}, 0)}{m-1} \quad (3.1)$$

$$P(S, K, j, r, \tau) = \frac{\sum_{i=1}^{m-1} S \times \max(\frac{e^{-r\tau}K}{S} - R_{i,j}, 0)}{m-1} \quad (3.2)$$

However, when attempting to determine the option price using (3.1) and (3.2), it quickly becomes obvious that the calculated price does not follow the put-call parity rule because the mean value μ of a real distribution does not equal zero (implying the real distribution is not risk-neutral). Notably, arbitraging opportunities occur when the distribution is not risk-neutral. Furthermore, the real distribution has its own volatility which is difficult to change. For example, if a real payoff distribution is formed based on a ten year period of sample data and has a standard deviation σ_1, but the forecasted volatility of the target option is σ_2, then the option must be priced using a distribution with a standard deviation σ_2 rather than σ_1. If the intrinsic volatility of the actual payoff distribution cannot be transformed to fit the short term volatility, the pricing error will be too large for practical use. Given the difficulty of changing the mean value without influence the variance, this study established a computational method for adjusting both the mean value and variance of an existing distribution to obtain the desired values while maintaining a similar distribution to the original.

To obtain risk-neutral characteristics based on the real distribution, the mean μ of the sampling data must be zero. By observing the real distribution, if the μ changes from a positive value to zero, the occurrence probability of rightmost (larger) sampling data reduces while the leftmost (smaller) sampling data increases. Based on this phenomenon, a computational method can be developed for adjusting the mean value of the real distributions by altering the sample occurrence possibilities.

The first step is attaching a weighting factor w_i to each sampled payoff rate $R_{i,j}$. Each w_i is assigned an original value 1.0, indicating that it has a "sampling count" of 1. The Call and Put prices thus can be represented as

$$C(S, K, j, r, \tau) = \frac{\sum_{i=1}^{m-1} S \times \max(w_i(R_{i,j} - \frac{e^{-r\tau}K}{S}), 0)}{m-1} \quad (3.1a)$$

$$P(S, K, j, r, \tau) = \frac{\sum_{i=1}^{m-1} S \times \max(w_i(\frac{e^{-r\tau}K}{S} - R_{i,j}), 0)}{m-1} \quad (3.2a)$$

For each set of sampling data, the mean value μ' and standard deviation σ' can be calculated as:

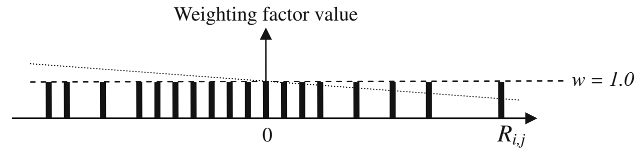


Fig. 1 Rotating the factor weights clockwise decreases the mean value of the distribution

$$\begin{cases} \mu' = \dfrac{\sum_{i=1}^{n} R_{i,j} \times w_i}{\sum_{i=1}^{n} w_i} \\ \sigma' = \sqrt{\dfrac{\sum_{i=1}^{n} (R_{i,j} - \mu')^2 \times w_i}{\sum_{i=1}^{n} w_i}} \end{cases} \text{for the jth tick to maturity}$$

The second step is to adjust the weighting factors to transform the real distribution into a risk-neutral manner. To achieve this, it is first necessary to sort the sampled payoff rates and position them on the X-axis with weighting factor 1. Assuming that the sample appearance probability changes linearly, the weighting factors can be rotated to modify the distribution, as illustrated in Fig. 1. Consequently, by fixing the rotation point to X = 0, the weighting factors can be rotated clockwise to decrease the mean values or anti-clockwise to increase them.

The weighting factors can be determined by solving the linear equations through the following steps:

$$\text{Let} \begin{array}{l} X_a = \sum_{i=1}^{n} R_{i,j} \big|_{R_{i,j} \geq 0} \\ X_b = \sum_{i=1}^{n} R_{i,j} \big|_{R_{i,j} < 0} \end{array} \text{and} \begin{array}{l} X_{a2} = \sum_{i=1}^{n} (R_{i,j})^2 \big|_{R_{i,j} \geq 0} \\ X_{b2} = \sum_{i=1}^{n} (R_{i,j})^2 \big|_{R_{i,j} < 0} \end{array} \quad (3.3)$$

Let m_a denote the slope of the weighting factors for $R_{i,j} \geq 0$, while m_b represents the slope of the weighting factors for $R_{i,j} < 0$.

$$\text{Solve} \begin{cases} X_a m_a = X_b m_b \\ \sum_{i=1}^{n} X_{i,j}(1 - m_a R_{i,j}) + \sum_{i=1}^{n} X_{i,j}(1 - m_b R_{i,j}) = 0 \end{cases}$$

$$\text{Then} \begin{cases} m_a = \frac{(X_a + X_b) X_b}{X_b X_{a2} - X_{b2} X_a} \\ m_b = \frac{(X_a + X_b) X_a}{X_b X_{a2} - X_{b2} X_a} \end{cases} \quad (3.4)$$

Thus, the weighting factor can be transformed as follows:

$$w_i = \begin{cases} 1 - m_a R_{i,j} \big|_{R_{ij} \geq 0} \\ 1 + m_b R_{i,j} \big|_{R_{i,j} < 0} \end{cases} \quad (3.5)$$

Combining (3.4) and (3.5) yields the following weighting formula:

$$w_i = \begin{cases} 1 - \frac{(X_a + X_b) X_b}{X_b X_{a2} - X_{b2} X_a} R_{i,j} \big|_{R_{i,j} \geq 0} \\ 1 + \frac{(X_a + X_b) X_a}{X_b X_{a2} - X_{b2} X_a} R_{i,j} \big|_{R_{i,j} < 0} \end{cases} \quad (3.6)$$

This computational method can transform any distribution into a risk-neutral distribution while largely preserving the characteristics of the original, as shown in Fig 2.

After transforming the real distribution into a risk-neutral distribution, the next step is to adjust its intrinsic volatility. If the intrinsic volatility after applying formula (3.6) is v, the forecast volatility is v'; formula (3.6) then can be rewritten as:

$$w_i = \begin{cases} \frac{v'}{v} \times \left(1 - \frac{(X_a+X_b)X_b}{X_b X_{a2} - X_{b2} X_a} R_{i,j}\right) \big|_{R_{i,j} \geq 0} \\ \frac{v'}{v} \times \left(1 + \frac{(X_a+X_b)X_a v'}{X_b X_{a2} - X_{b2} X_a v} R_{i,j}\right) \big|_{R_{i,j} < 0} \end{cases} \quad (3.7)$$

Notably, v' must be measured using the time to maturity scale (most option pricing applications use annual volatility). Supposing t days (t is a real number) to maturity and anticipated annual volatility is σ, v' can be estimated by:

$$v' = \sqrt{\frac{t \times \sigma^2}{365}} \quad (3.8)$$

Formula (3.7) can transform the real distribution into the desired volatility without affecting its mean value while maintaining a similar shape to the original distribution. Figure 3 shows the transformed distribution. The option price thus can be determined via (3.1a), (3.2a), (3.3), (3.7) and (3.8).

3.3 The algorithm

The full pricing algorithm comprises two parts. The first part is the algorithm for preparing the distribution map, while the second part is the pricing algorithm.

3.3.1 The real distribution generating algorithm

This algorithm is used to generate the real distribution map to accelerate the calculation process. Because the real distribution is repeatedly reused for the pricing algorithm, it is optimum to insert new sampling data into the existing distribution maps at the beginning of every trading day (or after trading hours). This algorithm requires minimal execution time if updates are daily performed. *SettlePrice* indicates the

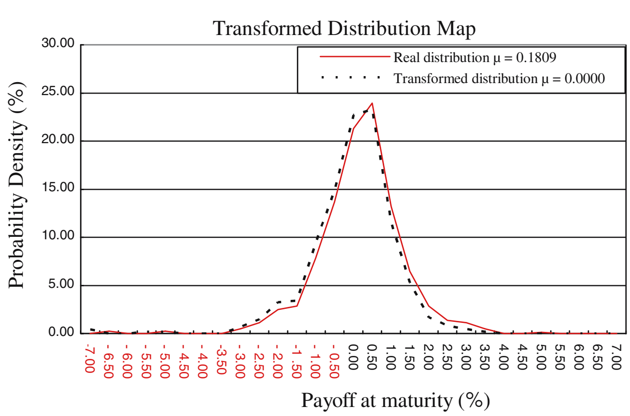


Fig. 2 The transformed distribution after rotated the weighting factors

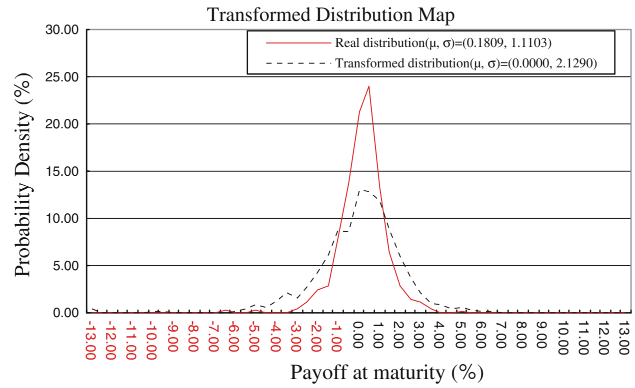


Fig. 3 Transformed payoff distribution after applying formula (3.7)

opening or closing price for the asset (depending on whether the final settlement price is determined based on the opening or closing price on the final settlement day) on the specified date *TransactionDate*. The sampling data for the previous day are gathered in a data set *{TimeTicks, TickPrice}* that contains the time tick count and tick price of the underlying asset. The results are stored in the *DistributionMap* table with the primary index set to *(Transaction_date, Time_Ticks)*. The *Transaction_date* field represents the sampling date, the *Time_Ticks* field indicates the time tick counts of the sample, and the *Return_Rate* field stores the asset return rate.

```
Algorithm MakeRealDistribution
  Input: SettlePrice, TransactionDate, {TimeTicks, TickPrice} of previous trading
         day
         Output: DistributionMap(Transaction_date, Time_Ticks, Return_Rate)
      Begin
         /* Clear old data to prevent duplication */
         DELETE FROM DistributionMap
            WHERE Transaction_date = TransactionDate
         /* Insert new data */
         For Each element pair in {TimeTicks, TickPrice}
            INSERT INTO DistributionMap (Transaction_date, Time_Ticks,
               Return_Rate)
               VALUES (TransactionDate, TimeTicks, SettlePrice/TickPrice)
         End For
      End Algorithm
```

In practical use of the *DistributionMap*, end users can also write their own programs to generate any desired mathematical distribution (or combinations) and store the generated samples into the *DistributionMap* table for the pricing algorithm to calculate the desired option price. For example, a researcher may use two lognormal distributions to simulate the twin-peak distribution as observed for the Taiwan stock market to verify whether it is worthwhile to apply two lognormal distributions to the BS model to improve the pricing performance. Researchers do not need to worry whether the two distribution combinations disobey the risk neutral characteristic before deriving sophisticated mathematical solutions. This characteristic increases the versatility of the pricing algorithm for cross field applications.

3.3.2 The pricing algorithm

This algorithm is used to price a European option with *DistributionMap* table generated by *MakeRealDistribution*. Suppose that the parameter set *(S, K, σ, T, r)* used to calculate the option price is *(SpotPrice, ExercisePrice, Volatility, TimeTicks, RisklessInterestRate)*, the pricing algorithm can be described as follows:

```
Algorithm GetOptionPrice
   Input: SpotPrice, ExercisePrice, Volatility, TimeTicks, RisklessInterestRate,
      TimeTicks
   Referenced Table: DistributionMap
   Output: CallValue, PutValue
Begin
   Define TargetRate = ExcPri/CrnPri – 1
   Define TargetMeanVaue = 0   //Suppose that the transformed distribution is
                               //Risk-Neutral

   SELECT Return_Rate, 1.0 as Weight
     FROM DistributionMap
     WHERE DistributionMap.Time_Ticks = TimeTicks
     INTO CURSOR TmpCursor ORDER BY Return_Rate ASC
                               //Generate Weighting Factors

  Let Cnt = record counts of TmpCursor

  Summation from TmpCursor
    Let A = ΣReturn_Rate for Return_Rate ≥ 0
    Let B = ΣReturn_Rate for Return_Rate <0
    Let A2 = Σ(Return_Rate ^2) for Return_Rate ≥ 0
    Let B2 = Σ(Return_Rate ^2) for Return_Rate < 0
   End Summation                //Formula (3.3)

   Let OriginalSD = the standard deviation of Weight in TmpCursor
   Let DaysToMarurity = transfer Timeticks to days to maturity

   Let TransformedVolatility = Square Root of (DaysToMaturity*Volatility^2)/365
                                //Formula (3.8)

   For Each record in TmpCursor
     Replace Weight With (TransformedVolatility/OriginalSD) * (1 – (( A + B) * B/(
       B * A2-A * B2)) * Return_Rate)For Return_Rate ≥ 0
     Replace Weight With (TransformedVolatility/OriginalSD) * (1 + (( A + B) * A/(
       B * A2 - A * B2)) * Return_Rate)For Return_Rate < 0
   End For                      //Formula (3.7)

   SELECT SUM(Weight * (SpotPrice * (1 + Return_Rate) - ExercisePrice))/Cnt
     FROM TmpCursor
     WHERE TmpCursor.Weight >= TargetRate
     INTO VARIABLE CallValue        //Formula (3.1a), processed by SQL

   SELECT SUM(Weight * (ExercisePrice - SpotPrice * (1 + Return_Rate)))/Cnt ;
     FROM TmpCur
     WHERE TmpCursor.Weight < TargetRate
     INTO VARIABLE PutValue         //Formula (3.2a), processed by SQL
   RETURN CallValue, PutValue
 End Algorithm
```

The above algorithm is carefully optimized for modern database applications involving SQL syntax and summarizing operations. The elimination of unnecessary floating point functions also increases the execution speed.

4 Empirical tests

This study uses tick price data for the period from 03/01/2001 to 17/12/2003 to verify the feasibility of using the proposed computational methods to price TAIEX options using real payoff distributions. There were 270 data recorded for each sampling day, and given the sample data set contained 216,810 entries. Data for the period 03/01/2001–31/12/2002 were adopted as the initial distribution map, and pricing errors in high frequency transactions were verified on the last trading day of each month during 2003. The trading hours of the TAIEX run from 9:00 to 13:00. The final settlement price was taken to be the opening price of the final settlement day. The verification procedure is presented below:

Step 1: Generate the initial distribution map
Filter out incorrect and duplicated data in the database, generate the distribution map using the *MakeRealDistribution* algorithm, and store it in a database table *DistributionMap (Transaction_date, Time_Ticks, Return_Rate)* that gives market price data on a per-minute basis between 03/01/2001 and 31/12/2002. Because the trading hours are 9:00 to 13:30, the first minute (9:01) is taken as *Time_Ticks* = 1 while the last (13:30) is *Time_Ticks* = 270. The *Return_Rate* $R_{i,j}$ equals the tick price of the TAIEX divided by the opening price for the following day:

$$R_{i,j} = \frac{X_{i+1,1}}{X_{i,j}}$$

Step 2: Determine the option price
This study uses an out-of-sample strategy to verify the pricing performance. The nearest three in-the-money and out-of-the-money call/put option prices were then calculated and priced using the *GetOptionPrice* algorithm for every time tick. The same option prices were also calculated using the BS model as a comparison. The riskless interest rate was the monthly fixed deposit interest rate used by the Central Bank of Taiwan.

Step 3: Estimate the pricing efficiency
The option price is the expected value of $S_t > K$ for a call option, or $S_t < K$ for a put option at maturity. Restated, for an ideal call price $C = \max(S_t - K, 0)$, the put price should be $P = \max(K - S_t, 0)$. Consequently, if an individual spends C dollars to purchase a call option, they should obtain C dollars by holding the option until maturity. The returning ratios *Rc* and *Rp* were calculated for each option price to determine the pricing efficiency where the ideal value is 1.0:

Table 2 Pricing error

	Computational method	Black-Scholes method
Call option, *Rc*	0.9290	0.9037
Pricing error	7.10%	9.63%
Put option, *Rp*	0.9874	0.9081
Pricing error	1.26%	9.19%

$$Rc = \frac{\sum \max(S_t - K, 0)}{\sum C} \text{for call options, and}$$

$$Rp = \frac{\sum \max(K - S_t, 0)}{\sum P} \text{for put options.}$$

Table 2 lists the final results. According to the empirical test, the computational method outperforms the traditional BS model in pricing performance.

Besides the pricing performance test, the execution speed was tested using Microsoft Visual FoxPro. The computational option model examined in this study is sufficiently efficient to price 1,000 option prices in 16 s (approximately 0.02 s each) where the distribution map contains 216,810 sample data, and is run on a 1GB RAM Intel Pentium4 2.6 GHz CPU personal computer system. All analytical results indicate that this computational method provides good pricing performance and efficient execution speeds when run on modern personal computer systems.

5 Conclusions

Most modern option pricing models apply mathematical distributions to approximate underlying asset behavior and attempt to calculate the desired option price using close form formulas. The empirical evidence based on observation of the actual payoff distribution suggests that the real distribution of a stock index is time variant and cannot be described using mathematical distributions, meaning the approach of most options pricing models is ineffective. To optimize the pricing performance, this study first introduces a computational model for pricing European options via real distributions and then demonstrates its practical feasibility using real world problems. This study solves two key issues in applying real distribution to options pricing. First, this study uses weighting factors to adjust the mean value of a real distribution to zero while maintaining its distribution characteristics in accordance with the put-call parity rule. Second, this study scales the distribution to adjust its standard deviation to meet the needs associated with applying dynamic volatility to practical problems. Solving these two issues makes this computational model highly suitable for cross field applications where mathematical distribution cannot be used to obtain feasible solutions, particularly for situations involving time variant distributions.

Although the proposed computational method is practical for real world application, room still exists for improvement. First, the weighting factor rotating method used to adjust the value of the distribution means can be enhanced. This study assumes linearly changing weighting factors. Nonlinear modification methodologies require further study. Second, this study uses a simple method based on adjusting standard deviation that may not be able to deal with complex applications. Third, the computational method must be simplified before it can be applied to execution speed critical applications.

References

Amin K, Jarrow R (1992) Pricing options on risky assets in a stochastic interest rate economy. Math Financ 2: 217–237

Bates DS (1991) The crash of ’87: was it expected? The evidence from options markets. J Financ 46:1009–1044

Bates D (1996) Jumps and stochastic volatility: exchange rate processes implicit in Deutsch mark options. Review of financial studies 9:69–107

Bates DS (1996) Jumps and stochastic volatility: exchange rate processes implicit in Deutsche mark options. Rev Financ Stud 9:69–107

Benaroch M (2002) Managing information technology investment risk: a real options perspective. J Manage Inform Syst 19(2):43–84

Black F, Scholes M (1973) The pricing of options and corporate liabilities. J Polit Econ 81(3):637–59

Carverhill AP, Cheuk THF (2003) Alternative neural network approach for option pricing and hedging. Available at SSRN: http://ssrn.com/abstract=480562 or DOI: 10.2139/ssrn.480562

Clemons EK (1991) Evaluating strategic investment in information system. Commun ACM 34(1):22–36

Cox JC, Ross SA (1976) The valuation of options for alternative stochastic processes. J Financ Econ 3: 145–166

Dos Santos BL (1991) Justifying Investment in new information technologies. J Manage Inform Syst 7(4):71–89

Hutchinson JM, Lo AW, Poggio T (1994) A nonparametric approach to pricing and hedging derivatives Securities Via Learning Networks. J Finance 49:851–889

Madan DB, Peter C, Eric CC (1998) The variance gamma process and options pricing. Euro Finance Rev 2(1):79–105

Meissner G, Kawano N (2001) Capturing the volatility smile of options on high-tech stocks: a combined GARCH-Neural network approach. J Econ Financ 25:276–292

Merton R (1973) Theory of rational option pricing. Bell J Econ 4: 141–183

Rubinstein M (1994) Implied binomial trees. J Financ 49:771–818

Scott LO (1997) Pricing stock options in a jump-diffusion model with stochastic volatility and interest rates: applications of fourier inversion methods. Mathe Financ 7: 345–358

Stoll HR (1969) The relationship between put and call option prices. J Finance 23: 801–824

Yacine A-S, Andrew WL (1996) Nonparametric estimation of state-price densities implicit in financial asset prices. J Finance 52: 499–548