Fast Generation of Implied Volatility Surface for Exchange-Traded Stock Options

Nan Zhang and Ka Lok Man

***Abstract*—We present an algorithm and its software implementation that computes implied volatilities for exchange-traded stock options. The LR (Leisen-Reimer) binomial tree is used for the underlying option pricing, which is adjusted for dollar cash dividends. The Brent's method is used as the root-finding procedure. The option pricing procedure that is at the core of the root-finding is optimised to maximise the performance. Tests were made on call and put options traded on the stocks of Microsoft Corporation and Apple Inc.. In 0.046 and 0.226 seconds, respectively, the implemented generator finished computing the implied volatilities for 154 Microsoft and 823 Apple call options.**

***Index Terms*—Implied volatility, volatility surface generation, LR binomial tree, Brent's method, option pricing**

I. Introduction

IN the BSM (Black-Scholes-Merton) [1], [2] option pricing model volatility is the standard deviation of the continuous compounding return of the underlying stock in one year's time. It is a measure of the uncertainty about the returns provided by the stock. In the settings of the original BSM model volatility of a stock is assumed to be constant during the lifetime of an option traded on the stock.

Given a stock and an option on the stock the volatility implied by the option price and the BSM pricing model can be computed using an iterative root-finding procedure. Contrary to the assumption made by the BSM model that the volatility is constant, the computed implied volatilities often vary with strike prices and expirations. In practice, different volatilities are used to price options having different strikes and expirations. The general pattern of the variation with strike in implied volatility for stock options is referred to as a volatility skew. The volatility used to price a low-strike option is higher than that used to price a high-strike option. The changing of implied volatilities with time to expiration is referred to as a volatility term structure. Combining volatility skews for different expirations generates a surface of implied volatilities that tabulates the volatilities appropriate as determined by the market for pricing options with certain strike and expiration.

Because option price is monotonically increasing in volatility, implied volatility computed from option price is often used as a proxy for option value. To compare the relative value of two options an investor needs only to look at their implied volatilities. As a proxy for option value, implied volatility presents the market's expectation of a stock's future price moves. A significant change in implied volatility means that there may be a shift in the expectation of the market towards a stock's future price. This provides to investors a tool for predicting the direction of a stock's future price moves.

Prices of exchange-traded stock options change in real time. The computation of the implied volatilities for exchange-traded options therefore should be performed with minimum delay. To this purpose we have developed a software that computes the implied volatilities from prices of exchange-traded options for different strikes and expirations. The underlying algorithm is able to handle multiple dollar cash dividends that take place within the lifetime of an option. To accelerate the computation the LR (Leisen-Reimer) [3] binomial tree is used for the underlying option pricing, because it has a much faster convergence speed than the commonly-used CRR (Cox-Ross-Rubinstein) [4] tree. Brent's method [5] is used as the root-finding algorithm which takes the pricing procedure at its core and searches for solutions on an iterative basis. Tests were made on an Intel quad-core 3.4GHz Corei7-2600. Using a single thread the software finished in 0.22 seconds generating implied volatilities for 823 stock options.

Organisation of the rest of the paper: Section II gives a brief background on exchange-traded stock options. Section III discusses the volatility computation algorithm in detail. This includes explanations on handling dollar cash dividends, the LR tree pricing method, the root-finding procedure and the optimisations to the option pricing procedure. Section IV discusses how common range of strike prices are selected for processing the options in the tests. Section V presents the test results. Section VI draws the conclusion.

II. Exchange-Traded stock options

Most stock options are traded on exchanges. Such exchanges in the United States include the Chicago Board Options Exchange (www.cboe.com) and NASDAQ OMX (www.nasdaqtrader.com). Exchange-traded options are American in style, which means they can be exercised at any time before or on the expiration date. For American options there is no closed-form pricing formula exist. Their prices must be computed using numerical procedures. In this work we use the binomial tree method.

Within the lifetime of many exchange-traded options cash dividends will be paid out by the underlying stocks. Because exchange-traded options are not adjusted for cash dividends we need to take the effect of cash dividends on option prices into consideration in the computation for implied volatilities. Information about exchange-traded options are easily accessible from the Internet. Fig. 1 shows a clipped screen

Manuscript received December 10, 2012; revised January 31, 2013. This work was supported by the XJTLU Research Development Fund under Grant 10-03-08.

Nan Zhang is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, China. Email: nan.zhang@xjtlu.edu.cn

Ka Lok Man is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, China.Email: ka.man@xjtlu.edu.cn

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Calls							Puts					
Price	Change	Bid	Ask	Volume	Open Int	Strike	Price	Change	Bid	Ask	Volume	Open Int
-	-	9.40	9.65	-	0	19.00	0.04	0.00	0.01	0.03	-	1176
-	-	8.50	8.70	-	0	20.00	0.04	0.00	-	0.02	-	2284
8.15	0.00	7.60	7.70	-	42	21.00	0.04	0.00	0.01	0.04	-	340
8.00	0.00	6.60	6.70	-	512	22.00	0.02	0.00	0.02	0.03	-	127
5.85	-0.40	5.60	5.70	41	329	23.00	0.03	0.00	0.03	0.04	20	398
4.94	-0.96	4.60	4.70	1	518	24.00	0.07	+0.03	0.06	0.07	101	1248

Fig. 1: Option chains on Microsoft's stock on July 13, 2012. All these options expire on August 17, 2012.

shot from Google Finance on vanilla call and put options traded on the stock of Microsoft Corporation. Exchange-traded options are often organised into chains. A chain consists of options with different strike prices but the same expiration date.

III. Computing implied volatilities

Computing the implied volatility of an option is to find the right value for the volatility parameter which if fed into an option pricing model will produce the option's traded price. The computation starts with an initial estimation for the volatility. This initial estimation is fed into the option pricing model to compute the option price under the estimation. This price is then compared with the option's traded price to calculate a closer estimation according to the difference. This procedure repeats until the option's price computed under an estimation for volatility becomes close enough to the option's traded price. At this point the estimation is treated as the implied volatility of the option.

A. Binomial pricing with cash dividends

Since exchange-traded stock options are American-style their prices must be computed by numerical procedures. For this purpose we use the binomial method. Most of exchange-traded options see within their lifetimes cash dividends paid out by their underlying stocks. For this reason we follow the method presented by John Hull [6] which incorporates dollar dividends into the binomial pricing. The advantage of this method is that it preserves the recombining structure of binomial trees.

The method calculates the present value of all cash dividends known to be paid out by the underlying stock within the lifetime of the option. An recombining binomial tree is then built where the root node corresponds to the stock's present price less the present value of all future cash dividends. After the whole tree is built, discounted cash dividends are added upon nodes of the tree at appropriate time steps. In what follows, we discuss this method in detail assuming a single dividend within the lifetime of the option. But this method can be easily generalised to handle multiple cash dividends.

Suppose D is the amount of cash dividend that is to be paid out at the ex-dividend time τ (measured in years). We use r to denote the annual continuously compound interest rate. The present value of the cash dividend is $De^{-r\tau}$. If S_0 denotes the present stock price, the uncertain component S_0^* of the price is $S_0^* = S_0 - De^{-r\tau}$. We then build a recombining binomial tree that models the dynamics of S^* – the uncertain component of the stock price process S. This tree is rooted at S_0^*. We use u and d to denote the up-move and down-move factors of the tree, respectively. On such a binomial tree a jth node at ith time step corresponds to the stock price $S_0^* u^j d^{i-j}$, where $j \in \{0, 1, 2, \ldots, i\}$. Note that at ith time step the number of nodes is $i+1$. Both indexes i and j start from zero. Now based on this tree that models S^* we convert it to another tree that models S. Since there is a single cash dividend D paid out at time τ we add D's discounted value onto all nodes whose time horizon proceeds τ. At time step i the nodes in the modified tree correspond to the stock prices $S_0^* u^j d^{i-j} + De^{-r(\tau - i\Delta t)}$, $j = 0, 1, 2, \ldots, i$ when $i\Delta t < \tau$, and $S_0^* u^j d^{i-j}$, $j = 0, 1, 2, \ldots, i$ when $i\Delta t > \tau$. The quantity Δt is the length of time represented by one step of the tree. If there are multiple cash dividends nodes at time step i need to be adjusted by the sum of the discounted values of all future dividends. Fig. 2 shows an example where there are two cash dividends within the lifetime of an option. The dashed lines show the tree that models S^*, and the solid lines show the modified tree that models S. The tree that models S^* is generated by the LR binomial tree method.

B. LR tree v.s. CRR tree

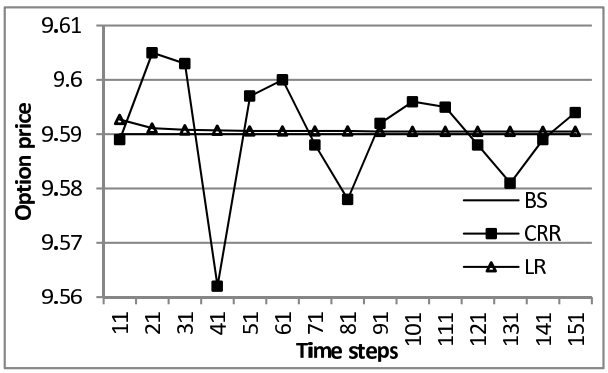


Fig. 3: Convergence comparison between the LR tree and the CRR tree using an European put option. The parameters were $S_0 = 40$, $K = 50$, $r = 0.1$, $\sigma = 0.4$ and $T = 0.5$.

In the computation for the implied volatilities the root-finding procedure may need to run many iterations. In each iteration the option price is evaluated under different estimations of the volatility. It is therefore very important to use an efficient option pricing procedure as it will be called many times during the computation. In our work we use the LR binomial tree [3] rather than the more common CRR tree [4]. The LR tree converges much faster than the CRR tree. Fig. 3 shows a comparison between the LR tree and the CRR tree using a deep in-the-money European put option as an example. It can be seen that the price computed by the LR tree converges smoothly to the BS (Black-Scholes) price of

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

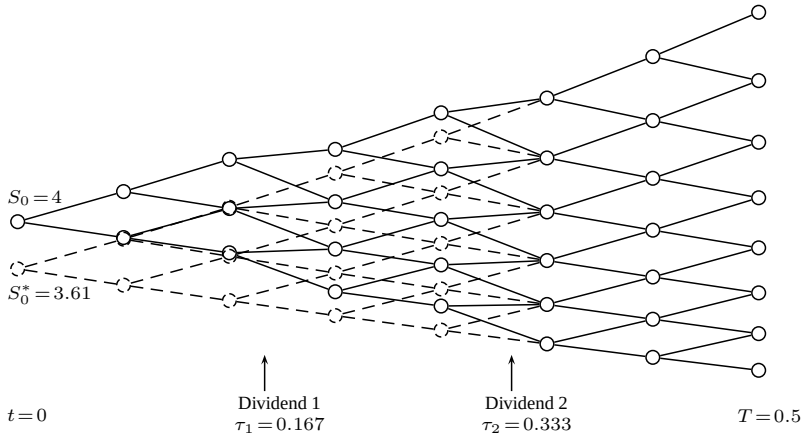


Fig. 2: Binomial trees for an American call with two dividends. The parameters were set as: current stock price $S_0 = 4$, strike price $K = 4$, interest rate $r = 0.1$, volatility $\sigma = 0.2$, expiration $T = 0.5$, time steps $N = 7$, ex-dividend times $\tau_1 = T/3$, $\tau_2 = 2T/3$ and cash dividends $D_1 = D_2 = 0.2$.

the option without the oscillation pattern demonstrated by the CRR tree. Note that the LR-tree method works only on odd numbers of time steps.

As before we denote the annual continuously compound interest rate by r, the option's expiration time by T, the number of time steps by N, the strike price by K and the uncertain component of the initial stock price by S_0^*. The up-move probability p, up-move factor u and down-move factor d used with the LR-tree method are set through the following equations.

$$
\begin{aligned}
d_1 &= \frac{\ln(S_0^*/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \\
d_2 &= \frac{\ln(S_0^*/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} \\
h^{-1}(z) &= \frac{1}{2} + \frac{\mathrm{sgn}(z)}{2}\sqrt{1 - \exp\left[-\left(\frac{z}{N + \frac{1}{3}}\right)^2\left(N + \frac{1}{6}\right)\right]} \\
\mathrm{sgn}(z) &= \begin{cases} 1 & z > 0 \\ 0 & z = 0 \\ -1 & z < 0 \end{cases} \\
p\prime &= h^{-1}(d_1) \\
p &= h^{-1}(d_2) \\
u &= \exp(rT/N)\frac{p\prime}{p} \\
d &= \frac{\exp(rT/N) - pu}{(1-p)}
\end{aligned}
\tag{1}
$$

C. The root-finding procedure

For the root-finding purpose we use Brent's method [5] which builds on an earlier algorithm proposed by Dekker [7].

D. Optimisations

To minimise the runtime required by the binomial pricing procedure we applied several source level optimisations to the code. A binomial tree conceptually is a two-dimensional structure. In main memory we store it in an one-dimensional array because explicit construction of a whole binomial tree is unnecessary. During the backward computation that starts from the leaf nodes the one-dimensional array stores only the option values represented by the nodes that are being processed. When the computation proceeds from the ith time step to the $(i-1)$th step the values represented by the nodes at the ith step are overwritten. This saves the time and space that are required by constructing a whole binomial tree.

On a binomial tree that models S^*, the uncertain component of the stock price, at the ith time step the nodes represent the stock prices $S_0^* u^j d^{i-j}$, $j = 0, 1, 2, \ldots, i$. But we did not use these expressions to compute the stock prices because using these expressions will cause lots of repetitive work. Instead, we only use this expression once when computing the stock price S_i^{*0} represented by the 0th node at the ith time step, and we have $S_i^{*0} = S_0^* d^i$ because $j = 0$. When computing the stock price S_i^{*1} we set it to be the product of S_i^{*0} and the constant u/d, and so $S_i^{*1} = S_i^{*0}(u/d) = S_i^* u d^{i-1}$, $j = 1$. When computing S_i^{*2} we set it to be $S_i^{*2} = S_i^{*1}(u/d) = S_i^* u^2 d^{i-2}$, $j = 2$. Each new stock price is obtained by multiplying the constant u/d onto the price that has just been computed. This avoids the repetitive evaluation of the power expressions u^j and d^{i-j}, and therefore significantly shortens the runtime of the binomial pricing procedure.

The same binomial pricing procedure in our program works both for European and American options and for call and put options. But we did not use branching statements to distinguish these situations. This is to avoid the penalty brought about by the potential mis-predication on the branching statements. To distinguish European and American options a lookup table of two positions was used. The first position of the table always stores zero, and the second stores the payoff from an immediate exercise. In the case of an European option an expected value is compared with the first element of the table, and in the case of an American option it is compared with the second. These two situations are unified by using a lookup index whose value is 0 for European options and 1 for American. To unify call and put options we calculate their payoffs by the formula $\max(I(S-K), 0)$, where I is 1 for calls and -1 for puts, and S and K are the stock and strike prices, respectively.

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Implementation

The programs in our work were written in C/C++. The compiled generator takes as input a text file that lists information about a stock, the option chains based on the stock and interest rate term structure. The output of the generator is a text file contains the computed implied volatilities arranged in a tabular form with columns being the expiration dates and rows the strike prices.

Options offered by the exchanges are organised into chains for different expiration dates. For a particular expiration date the call or put option chains contains a series of options. With all other parameters being the same the options in a chain differ in their strike prices. In most cases, the increment in strike prices between each pair of successive options is a fixed amount, which depends on the present price of the stock. To build an implied volatility surface for a stock that includes all the options in all the chains we need to choose common strike prices for all the options. However, for exchange-traded options the range of strike prices are often different in different chains that expire on different dates. Moreover, the strike price increment in different chains can be different as well. All these irregularities pose problems for the volatility computation.

To solve these problems we choose the minimum and maximum strike prices that are found in all the option chains on a stock as the lower and upper bounds of the range of strike prices. The increment between each pair of strikes we choose is a value that will include most of the options in all the chains in the computation. In choosing the range of the strikes and the increment call and put option chains are dealt separately. Once a range of strikes and the increment have been chosen the generator will go through this range starting from the smallest strike price. For each strike price in the range and an option chain the generator will find the option with that strike price in the chain and compute the volatility implied by its exchange-traded price. The generator repeats this process until all the option chains listed in the input text file have been processed. If a strike price in the range is not found in an option chain the generator will by-pass this strike price and proceed to the next. In the output text file the generator will write a special symbol in the entry that corresponds to the un-found strike price. If in any option chain, an option's strike price is not found in the range the option will be ignored by the generator and no information will be output for that option. This can happen if in some option chain the strike prices follow an irregular increment pattern.

V. Tests

We made tests using options traded on the stock of Microsoft Corporation (stock code MSFT) and Apple Inc. (stock code AAPL). The information about the options were collected on May 23, 2012, at which time Microsoft's stock was traded at $29.11 per share and Apple's stock at $570.56 per share. For each of the two stocks we selected 6 call option chains and 6 put option chains. The options in these chains expired in 23, 58, 86, 121, 149 and 240 days, respectively. These were converted to years when implied volatilities were computed, assuming 365 days a year. The interest rates corresponded to each of the expiration dates were 0.2397%, 0.3458%, 0.4588%, 0.5754%, 0.6525% and 0.8559%. The range of strike price for the Microsoft options were minimum $13, maximum $47 and increment $1, and the range for the Apple options were $135, $960 and $5, respectively. At the time the data were collected no dividend information was available for Apple's stock, but Microsoft would pay out $0.2 at two future dates within the lifetime of the options that had the longest expiration. The distances of the two days to May 23, 2012 were 0.231 and 0.481 years, respectively.

The tests were made on an Intel quad-core 3.4GHz Corei7-2600 processor under Ubuntu Linux 10.10 64-bit version. The programs were compiled by Intel C/C++ compiler icpc 12.0 for Linux with -O3 and -ipo optimisations. The compiled generator finished in 0.046 seconds in computing the implied volatilities for the 154 Microsoft call options, and in 0.045 seconds for the same number of Microsoft put options. The runtimes in processing the 823 Apple call and 823 Apple put options were 0.226 and 0.222 seconds, respectively. The timing results were obtained by running the generator using a single thread. In all the computations the number of time steps was fixed to 121 in the LR binomial pricing.

From the computed implied volatilities we generated plots. Fig. 4 shows the plots for the volatility skews (4(a)-(d)) and the volatility term structure (4(e)-(h)). Fig. 5 shows the surfaces which combine the volatility skew and term structure.

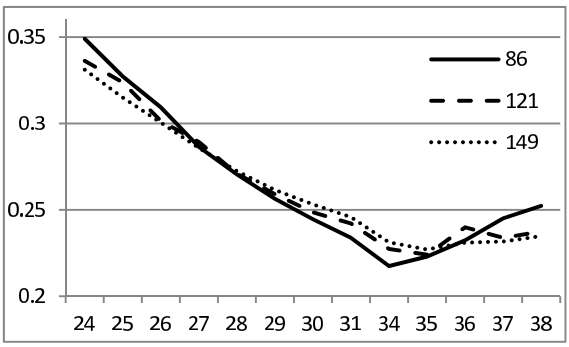
Volatility skew describes the relationship between implied volatility and strike price. The typical pattern for volatility skew, as it is shown by the curves in Fig. 4(c), is that implied volatility decreases as strike price increases. This means that the volatilities used to price options of low strike prices (i.e., deep in-the-money calls and deep out-of-the-money puts) are higher than that used to price options of high strike prices (i.e., deep out-of-the-money calls and deep in-the-money puts). Three curves are plotted in each of the figures 4(a)-(d). They are for options expiring in 86, 121 and 149 days, respectively.

The volatility term structure describes the relationship between implied volatility and time to expiration. Implied volatility tends to be increasing as time to expiration increases when short-dated volatilities are historically low. This is the situation demonstrated by the curves in figures 4(g)-(h). Similarly, implied volatility tends to be decreasing as time to expiration increases when short-dated volatilities are historically high. This seems to be the situation described by the solid-line curve in Fig. 4(e). The curves plotted in each of the figures 4(e)-(h) are for options with different strike prices.

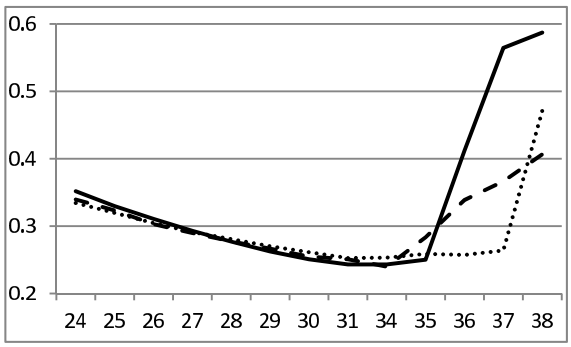
VI. Conclusion

We have presented an implied volatility generator that computes implied volatilities for exchange-traded call and put options. We use the binomial method as the underlying option pricing model and implemented it using the LR trees. The pricing process handles multiple dollar cash dividends by separating a stock's price into a certain component and an uncertain component. The certain component consists of the discounted value of all cash dividends. The uncertain component is the stock's price less this certain component. To price an option of the stock a LR tree is first built using

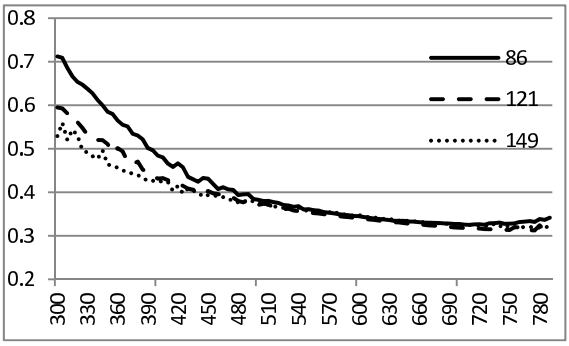
ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



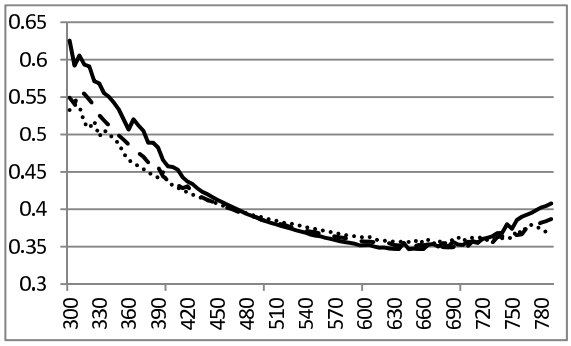
(a) MSFT call volatility skew for different expirations.



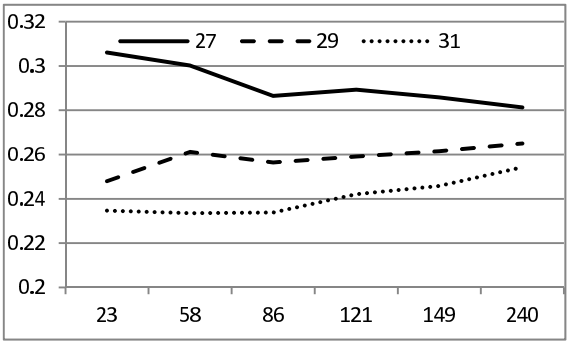
(b) MSFT put volatility skew for different expirations.



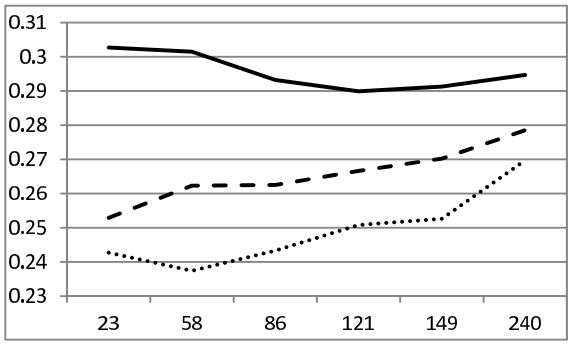
(c) AAPL call volatility skew for different expirations.



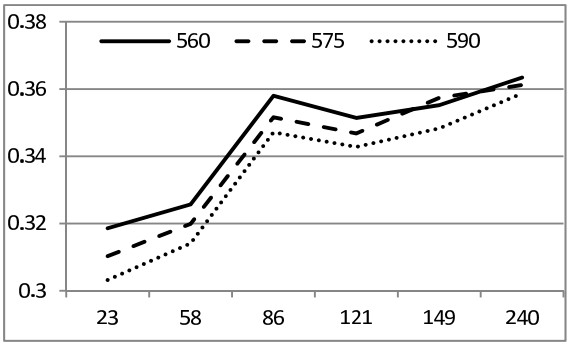
(d) AAPL put volatility skew for different expirations.



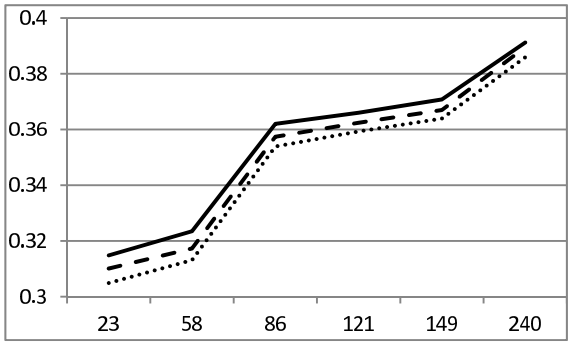
(e) MSFT call volatility term structure for different strikes.



(f) MSFT put volatility term structure for different strikes.



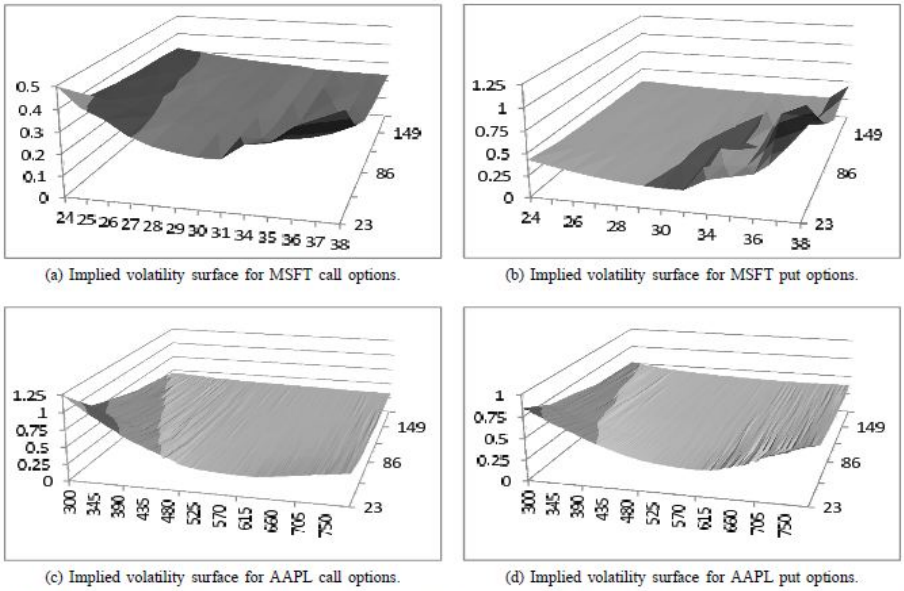
(g) AAPL call volatility term structure for different strikes.



(h) AAPL put volatility term structure for different strikes.

Fig. 4: Volatility skew and term structure for MSFT and AAPL call and put options. The x-axes in (a)-(d) are labelled by strike price, and in (e)-(h) are labelled by expiration date.

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) Implied volatility surface for MSFT call options.

(b) Implied volatility surface for MSFT put options.

(c) Implied volatility surface for AAPL call options.

(d) Implied volatility surface for AAPL put options.

Fig. 5: Implied volatility surface for MSFT and AAPL call and put options. The x-axes are labelled by strike, y-axes by expiration and z-axes by implied volatility.

the present value of the uncertain component as root and then the discounted value of the dividends are added upon nodes of the tree at appropriate time steps. The Brent's algorithm was used as the root-finding method.

Source code level optimisation techniques were applied to the option pricing procedure to minimise its runtime. Tests were made on options traded on stocks of Microsoft Corporation and Apple Inc.. From the computed results implied volatility skew, term structure and surface were plotted and presented. Some of the plots confirm to the well-known pattern for volatility skew or term structure.

REFERENCES

[1] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *The Journal of Political Economy*, vol. 81, no. 3, pp. 637–659, 1973.

[2] R. Merton, "Theory of Rational Option Pricing," *Bell Journal of Economics and Management Science*, vol. 4, pp. 141–183, 1973.

[3] D. Leisen and M. Reimer, "Binomial Models for Option Valuation - Examining and Improving Convergence," *Applied Mathematical Finance*, vol. 3, pp. 319–346, 1996.

[4] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option Pricing: A Simplified Approach," *Journal of Financial Economics*, vol. 7, no. 3, pp. 229–263, 1979.

[5] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Englewood Cliffs, NJ, 1973, ch. 4.

[6] J. C. Hull, *Options, Futures, and Other Derivatives*, 8th ed. Prentice Hall, 2012, ch. 20.3.

[7] T. J. Dekker, "Finding A Zero by Means of Successive Linear Interpolation," in *Constructive Aspects of the Fundamental Theorem of Algebra*, B. Dejon and P. Henrici, Eds. Wiley-Interscience, London, 1969.

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)