# Report on the Paper titled “Calibrating Option Pricing Models with Heuristics”

Mutian Liu

*Undergraduate at the University of Michigan*
*lmutian@umich.edu*

**ABSTRACT**
This paper was based on the fact that calibrating option pricing models to market prices usually result in optimization issues to which standard strategies (as some based on gradients) cannot be used. It investigated two different models: Bates’s model, and Heston’s stochastic volatility model, and they both include jumps. It discusses how to price options in these models, as well as how to calibrate the parameters of the models with heuristic techniques.

***Keywords**: calibrating option pricing, optimization, Bates’s model, Heston’s model, heuristic*

## 1. INTRODUCTION

In finance, the success of the BSM model stems less from its empirical quality than from its computational convenience. This convenience comes in two flavors. Firstly, we have closed-form pricing equations (Gaussian distribution functions cannot be analyzed, but fast and accurate approximations exist). Secondly, the calibration model requires the identification of only one parameter, volatility, which can be easily calculated from market prices using Newtonian methods or other zero-finding techniques. With the Heston and Bates models, both of these tasks become more difficult. Pricing requires numerical integration and calibration requires finding five and eight parameters, whereas for the BSM there is only one parameter.

The authors will examine the calibration of these models. Finding the parameters that align the models with market prices implies solving a non-convex optimization problem. The authors suggest the use of optimization heuristics, and more specifically they show that both differential evolution and particle swarm optimization provide good solutions to this problem.

## 2. LITERATURE REVIEW

The use of mathematical methods to analyze financial problems dates back to the early 20th century. In 1900 the French mathematician Bachelier published his doctoral thesis 'Theory of Speculation'. He argued that in the capital market there are buyers and sellers, that buyers are bullish and sellers are bearish, that the price fluctuations are Brownian movements and that their statistical distribution is normal, which gradually led to the subsequent theory of the capital market[1]. After 50 years of silence, Bachelier's work was rediscovered. In 1952, Markowitz published "Portfolio Selection", which revealed for the first time from the perspective of mathematical economics how to choose the optimal portfolio through the efficient frontier of the portfolio and how to reduce risk through diversification, pioneering modern investment analysis theory[2]. In 1963, Sharpe, a student of Markowitz, proposed a simplified form of the calculation, now known as the single exponential model, which made the base of the theory for investment[3]. Johnson and Stein used the bond portfolio approach to extend portfolio theory to hedging. In the late 1970s, Ederington developed Johnson and Stein's work to include financial futures to hedge financial price risk, which led to modern hedging theory.[4]

In 1958, Modigliani and Miller, in their study of the relationship between corporate capital structure and corporate value, published an important and epoch-making result, the so-called MM theory, which implies an extremely deep. This theory contains a profound idea, the idea of no-arbitrage equilibrium, which later had a great impact and became an important analytical tool in a series of subsequent financial research results, such as arbitrage pricing theory and option pricing theory. It has also become a fundamental analytical technique in today's financial engineering for product design, development and facilities.[5]

In 1973, Brennen relaxed the no-tax assumptions of the capital asset pricing model to take into account the effect of tax rates[6]. In 1965, Fama and Samuelson proposed the Efficient Market Theory, which states that in a properly functioning capital market, the evolution of asset prices can be described by a lower harness process and that the best estimate of the value of a bond is today's price.[7]. In 1973, Black and Scholes proposed the first complete model of option pricing, the Black-Scholes formula[8], which was widely accepted and applied by both the theoretical and practical communities and became another revolution in the field of finance. In the same year, Merton gave a model with continuous branch interest rates and coefficients are not constant in the important case, thus refining the theory of option pricing.

In 1976, Ross further developed the capital asset pricing model and introduced the theory of arbitrage pricing. The theory does not require the same stringent assumptions as the capital asset pricing model and its model takes the same form as the multi-index model[9]. For mathematical finance, one of the main objects of study is the pricing, modeling and hedging problems of marketable securities and their derivatives, such as forward contracts, futures, options and swaps. The main problem to be solved in the study of derivative securities is how to determine the price of the derivative security, i.e. the pricing of the derivative security, and the other is the hedging of the derivative security. Among all of the derivative securities, options are the most widely studied.

## 3. METHODOLOGY

### 3.1. Pricing with the characteristic function

The nature of the BSM is a no-arbitrage argument; it generates a partial differential equation which can be solved numerically or analytically. A recent method is based on the characteristic function of (log) stock prices. Euclidean options can be priced by the equation below:

$$C_0 = e^{q\tau} S_0 \Pi_1 - e^{-r\tau} X \Pi_2 \tag{1}$$

$$\Pi_1 = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left(\frac{e^{-iw\log(X)}\Phi(\omega - i)}{i\omega\Phi(-i)}\right) d\omega \tag{2}$$

$$\Pi_2 = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left(\frac{e^{-iw\log(X)}\Phi(\omega)}{i\omega}\right) d\omega \tag{3}$$

It defines $\Pi_j^* \equiv \pi\left(\Pi_j - \frac{1}{2}\right)$ as the integrals in the equations. The marker $\Phi$ is the characteristic function of the log stock price; the function Re (·) gives the actual part of a complex number. For a certain $\Phi$ it can compute $\Pi_1$ and $\Pi_2$ by numerical integration, and therefore acquire option prices.

Under a BSM model, the stock price $S_t$ in the risk-neutral measure follows:

$$dS_t = (r - q) S_t d_t + \sqrt{v} S_t dz_t \tag{4}$$

$$T_0 = e^{-q\tau} S_0 N(d_1) - X e^{-r\tau} N(d_2) \tag{5}$$

The above equation is the well-known pricing formula for the BSM call.

Given the dynamics of S, the log price s τ = log ( S τ ) comply with Gaussian distribution where $S_0$ is the natural logarithm of the current spot price. The characteristic function of s τ is given by

$$\Phi_{BSM}(\omega) = E\left(^{i\omega s_\tau}\right) = \exp(i\omega s_0 + i\omega\tau(r - q) - 1/2(i\omega + \omega^2)\tau v) \tag{6}$$

Merton (1976) proposed to model the underlier's movements as a diffusion with discontinuous jumps; therefore acquire

$$dS_t = (r - q - \lambda\mu_J) S_t dt + \sqrt{v} S_t dz_t + J_t S_t dN_t \tag{7}$$

$N_t$ is a poisson counting process, with intensity $\lambda$; $J_t$ is the random jump size (if a jump happened). Under Merton's model the log-jumps are distributed as

$$\log(1 + J_t) \sim N\left(\log(1 + \mu_J) - \frac{\sigma_j^2}{2}, \sigma_j^2\right) \tag{8}$$

The pricing equation is defined as this (Merton,1976, P.135):

$$C_0 = \sum \frac{e^{-\lambda\tau'}(\lambda'\tau)^n}{n!} C_0'\left(r_{n'} \sqrt{v_n}\right) \tag{9}$$

where $\lambda' = \lambda(1 + \mu_J)$ and $T_0'$ is the BSM equation (5), although the prime shows that $T_0'$ is deemed at adjusted values of $r$ and $v$.

Factorials will easily result in an overflow (Inf), which is benign for two reasons. First, it does not need big numbers for n, a value of around 20 is well-sufficient. Second(if we insist on large n), software packages as Matlab or R will deem 1/Inf as zero, therefore the summing will add zeros for large n.(Numerical analysts prefer to replace n! by $\exp\left(\sum_{i=1}^n \log i\right)$ as this result in better accuracy for large n. For Merton's model it is not required.) In light of the implementation, working with big values for n will still result in a warning or an error, and so disrupt a calculation. In R for example, the handling of such a warning will rely on the options setting:

```
> options()$warn
[1] 0
```

The above is the standardized setting. Calculating the factorial for a big number will lead to a warning;

```
> factorial(200)
[11 Tnf
Warning message:
In factorial(200): value out of range in 'gammafn'
```

However, with warn set to 2, any warning are to be changed into an error. Thus:

```
> options(warn=2)
> Error in factorial(200):
(converted from warning) value out of range in
'gammafn’
```

our calculation disrupt. It needs to safeguard against these possible errors: it can for example substitute the function call factorial(n) by its real calculation which generates:

```
> options(warn=2)
> exp(sum(log(1:200)))
[1] Inf
> Prod(1:200)
[1] Inf
```

In the Heston (1993) model the stock price S and its variance $v$ are defined as

$$dS_t = rS_t d_t + \sqrt{v_t} S_t dz_t \tag{10}$$

$$dv_t = k(\theta - v_t) d_t + \sigma \sqrt{v_t} dz_t \tag{11}$$

The mean reversion speed is $k$ and $\sigma$ is the volatility-of-volatility, long-term variance is defined as $\theta$, the Wiener processes $z^{(.)}$ have correlation $\rho$. For $\sigma \to 0$, the Heston dynamics approach those of BSM. A thorough analysis of the model will be traced in Gatheral’s research[10]. The characteristic function of the log-price in the Heston model was defined as below, see Albrecher’s research[11].

$$\phi_{Heston} = e^{A+B+C} \tag{12}$$

With only 5 parameters(in the risk-neutral probability), the Heston model can produce a volatility smile.

In the Bates model, described in Bates’ research[12], makes jumps to the dynamics of the Heston model. The stock price S and its variance v are defined as

$$dS_t = (r - q - \lambda\mu_J) S_t dt + \sqrt{v_t} S_t dz_t + J_t S_t dN_t \tag{13}$$

$$dv_t = k(\theta - v_t) dt + \sigma \sqrt{v_t} dz \tag{14}$$

$N_t$ is poisson count process with intensity $\lambda$, therefore the possibility a jump of size one occur is $\lambda dt$. As ubder Merton's model, the logarithm of the jump size $J_t$ is distributed as a Gaussian.

We use the Gauss-Legendre rule, see Davis and Rabinowitz[13], Trefethen[14]and Appendix; we have also experimented with other schemes such as Gauss-Lobatto[15], but given the accuracy required for our problem (option prices do not need to be computed to 8 decimal places), no integration scheme significantly better than the other schemes. Therefore, in the following, instead of using the equations, we calculate the nodes and weights and evaluate the integrals directly.

In order to test the pricing procedure, we need to analyze the BSM model and the Merton jump diffusion model. For the above models, we can compare the solutions obtained by the classical formulation with those obtained by integration. Furthermore, we can explore several extreme cases: for Heston with zero volatility, we should obtain the BSM value; for Bates with zero volatility, we should obtain the same value as in the Merton jump diffusion. (Of course, Bates with zero volatility and no jumps should still obtain the BSM value.)

### 3.2. Calibrating model parameters

Calibrating an option pricing model means finding parameters to match the model price to the market price, which gives rise to an optimization problem of the following form

$$\min \sum_{i=1}^{M} \frac{\left| C_i^{\mathrm{mod}el} - C_i^{market} \right|}{C_i^{market}} \tag{15}$$

where M is the number of market prices. You can also specify absolute deviations, replace absolute values with squared values or introduce a weighting scheme. The choice of the objective function depends on the scope of the application, and ultimately the determination of a good objective function is an empirical question. In this case, we are interested in the numerical aspect and therefore use the specification (14). Figure 1 displays a objective function case for the Heston model (on a log scale) when varying two parameters - mean reversion $k$ and volatility-of-volatility $\sigma -$ when holding the others constant. The problem is not convex, so standard methods (e.g. derivatives based on the objective function) may not work. We apply heuristics such as differential evolution and particle swarm optimization to solve (14).

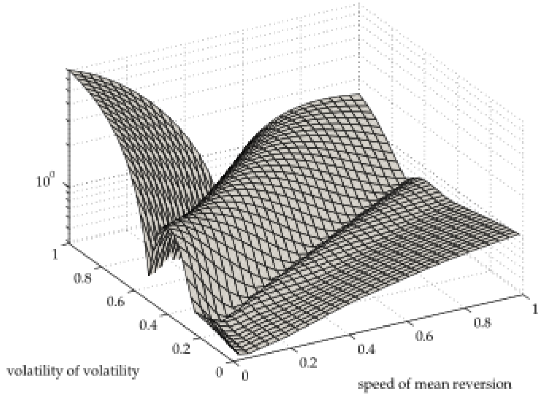

**Figure 1** A search space for the Heston model

In evaluating (14), instead of determining the price of a single option, we determine the price of a whole set of options with different maturities. However, for a given set of parameters describing the underlying process of the model, the characteristic function depends only on the time to maturity and not on the strike price. This suggests that speed can be improved by pre-processing the members of p that are constant for a particular maturity and then calculating all strike prices for that maturity, see Kilin[16]for a discussion, summarised in Algorithm 1.

Differential Evolution (DE) is described in detail by Storn and Price[17]: Differential Evolution has np solutions, stored in a real vector of length p (p is 5 for Heston and 8). In each generation k, the algorithm creates a new candidate solution $n_p$ , one new solution for each existing solution. Such a candidate solution is created by taking the difference between the other two solutions, weighting this difference, and adding the weighted difference to the third solution. An element-by-element crossover is then performed between the auxiliary solution $P^{(v)}$ and the original solution. If such a final candidate is better than the original solution, it is replaced by the original solution, otherwise the original solution is retained.

In particle swarm optimization[18], there is again a population of $n_p$ solutions stored in real-valued vectors. At each generation, the solutions are updated by adding another vector $v_i$ (called velocity). A solution is a position in the search space and the velocity can be seen as the direction in which the solution is moving. During the optimization process, the velocity changes. The magnitude of the change is the sum of the two parts. The magnitude of the change is the sum of two parts: the movement of a particular solution in the direction of the best solution it has found so far, $Pbest_i$ and the movement in the direction of the best solution for the whole group, $Pbest_i$ . These two directions are perturbed and summed by multiplying by a uniform random number $\zeta$ and a constant $c_{(.)}$ . The resulting vector is added to the previous solution and the resulting update rate is added to the corresponding solution. In some embodiments, the speed is reduced in each generation by setting a parameter $\delta$ called inertia to a value less than 1.

Initial tests showed that the objective function tends to be flat, so that different parameter values produce similar objective function values. This suggests that our problem may be sensitive to small changes in the data when we are interested in exact parameter estimates, and if we insist on exact parameter calculations, we may need multiple iterations or an algorithm with large step sizes. Therefore, as a local search strategy, we use the direct search method of Nelder and Mead[19], implemented in Matlab's fminsearch. This algorithm can vary its step size, and it is also robust in the presence of noisy objective functions. Algorithm 4 summarized the hybrid search.

Algorithm 4 Hybrid search.

```
set parameters for population-based method
for k =1 to nc do
    do population-based search
    if local search then
        select ns solutions as starting values for local search
        for each selected solution do
            perform local search
        end for
    end if
end for
```

During implementation it is necessary to decide how often to start a local search, how many solutions to choose and how to select them. In an extreme case, the strategy would be a simple one of using only one generation and restarting the local search method.

Spendley et al.[20] proposed encoding a solution x as a mere x. A single line of dimension p consists of p + 1 vertices (points), so that it is a line segment for p = 1, a triangle for p = 2, a tetrahedron for p = 3, and so on. In the algorithm of Spendley et al. (1962), this singular can be reflected through an edge or collapsed. Thus, the size of the monoclinic can change, but not its shape. Nelder and Mead added two more operations, namely that the monogram can be stretched and contracted, and therefore the monogram can change its size and shape.

In Algorithm 5, we describe the Niederer-Mead algorithm. The notation follows Wright[21] and the solution is organised as follows.

$$x_1, x_2, \cdots x_{p+1}$$

that means we have

$$F(x_1), F(x_2) < \cdots F(x_{p+1})$$

Define the objective values connected with particular solutions as $F_1, F_2, \cdots F_{p+1}$.the values for the parameters in Algorithm 5 are

$$\rho = 1, \chi = 2, \gamma = 1/2, and \varsigma = 1/2$$

they are as well applied in Matlab's fmingearch. Matlab transforms the starting guess $x$ into

$$\begin{array}{ccccc} x^{(1)} & x^{(1)} + \varepsilon x^{(1)} & x^{(1)} & x^{(1)} & \cdots x^{(1)} \\ x^{(2)} & x^{(2)} & x^{(2)} + \varepsilon x^{(2)} & x^{(2)} & \cdots x^{(2)} \\ x^{(3)} & x^{(3)} & x^{(3)} & x^{(3)} + \varepsilon x^{(3)} & \cdots x^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots \vdots \\ x^{(p)} & x^{(p)} & x^{(p)} & x^{(p)} & \cdots x^{(p)} + \varepsilon x^{(p)} \end{array}$$

where the superscript (i) implied the i-th element of x. In the implementation applied here(Matlab 2008a), $\varepsilon$ is 0.05. If $x^{(i)}$ is 0, then $\varepsilon x^{(i)}$ is set to 0.00025.

Simplex adapts to the contours of the objective function (i.e. it can 'stretch'), allowing it to take large steps in favorable directions. However, this flexibility can also be a drawback. For example, imagine a narrow valley through which a long, stretched monolith can pass. If this valley curves, the monad will not easily fit. This phenomenon also seems to be a problem for us. When the monad is initialised, the maximum parameter value is 5% larger than its minimum, and structurally this is true for all parameters, see equation (14). Thus, the relative growth along any dimension is uniform. If we search and compare this initial singular with the final singular, we find that it grows by a factor of over 200 in just one dimension, and that the number of terms in the singular often increases from around 10 to over 1012. It turns out that restarting the algorithm, i.e. initializing the singletons several times, leads to a better solution.

We therefore expect non-negative variance, correlation between -1 and 1 and parameters such as $k$ , $\sigma$ and $\lambda$ to be non-negative as well. These constraints are achieved through penalty clauses. If there is a violation, a positive number proportional to the violation is added to the objective function.

## 4. RESULTS

To test the author's technique, artificial data set was created. The spot price $S_0$ was 100, the risk-free rate $r$ was 2% and there was no return. We calculated prices for two strike levels X, from 80 to 120, as well as 1/12, 3/12, 6/12, 9/12, 1, 2 and 3 year maturities $\tau$ . The system therefore consists of 21 x 7 = 147 prices. Given the parameters, we calculate the option prices and record them as actual prices. We then run each method 10 times to solve (13) and see if we can recover the parameters. This setup implies that a perfect fit is possible.

The parameters for the Heston model are taken from the table below.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\sqrt{v_0}$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.4 | 0.2 | 0.5 | 0.6 | 0.6 | 0.7 | 0.8 |
| $\sqrt{\theta}$ | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.4 | 0.5 | 0.3 | 0.3 | 0.3 | 0.3 |
| $\rho$ | −0.3 | −0.7 | −0.9 | 0.0 | −0.5 | −0.5 | 0.0 | −0.5 | −0.5 | −0.5 | −0.5 |
| $\kappa$ | 2.0 | 0.2 | 3.0 | 3.0 | 0.2 | 0.2 | 0.5 | 3.0 | 3.0 | 2.0 | 1.0 |
| $\sigma$ | 1.5 | 1.0 | 0.5 | 0.5 | 0.8 | 0.8 | 3.0 | 1.0 | 1.0 | 1.0 | 1.0 |

The Bates model uses parameter sets.

Each model has 10 different parameter sets, 10 optimization runs (restarts) for each parameter set, and 10 outputs for each optimization method. For each restart we store the value of the objective function and the corresponding parameter estimates. For the latter, we calculate the absolute error, i.e.

Error = | estimated parameters - true parameters |.

We then analyze the distribution of these errors.

All algorithms are coded in Matlab. We use fminab for direct search. In DE the parameter F should be set around 0.3-0.5 (we use 0.5). In RS the main goal is to speed up convergence. It is not good to accelerate too much, so we keep the inertia below unity level (here o.7) and limit the absolute value of acceleration to o.2. The stopping criterion for DE and ps is a fixed number of function estimates (population size x number of generations) and 3 settings.

Using an Intel 87oo processor with a 2.53 GHz core and 2 GB of RAM, a single run takes about 10, 40 and 160 seconds, respectively. Another stopping criterion is to stop the algorithm when the diversity of the population (as measured by the objective function or a set of parameter values) falls below an acceptable level. This strategy works well for DE, since its solution converges quickly, but increases the runtime of ps).

The hybrid method is run for 50 generations with a population of 25 solutions. Every tenth generation, one or three solutions are selected that are either the best ("elite") solutions or random solutions. These solutions are used as starting values for the local search. This search is performed under the Nelder-Meade constraint for up to 200 iterations until further improvements are achieved; each run takes between 10 and 30 seconds.

## 5. DISCUSSIONS

For the purely population-based approach, the gray scale shows increasing estimates of the function (1,250, 5,000, 2,000).

For the Heston model, all methods converged quickly to a very good solution with a pricing error of less than 1% and were perfectly consistent (although not reported, additional DE and ps tests were run to increase the estimate of the function and extend the run time to 3-5 minutes, but both algorithms (without exception) converged to a true solution). However, for practical purposes, this level of accuracy was sufficient. DE converges faster than PS. This is evident in the hybrid algorithm. Here, in the case of DE, it does not matter how the solution for the local search (random or elite) is chosen, as all members of the population are similar to each other; in the case of PS, it is more efficient to choose the solution by quality.

The calibration of the Bates model is a more difficult task. The results showed that convergence is slower here. The hybrid approach is very efficient, especially when both DE- and rs-based solutions are chosen: for the Heston and Bates models, the hybrid algorithm works best, with a slight advantage for the DE-based algorithm.

According to the results, in the case of the Heston model the parameters converge almost as fast as the objective function.

For the Bates model, the results are even worse. The parameters, also included in the Heston model, are not estimated very accurately, but for the jump parameters (A, yy and o), there is almost no convergence. The fact that the parameters do not converge does not mean that a good fit cannot be achieved. The lack of convergence can also be attributed to the choice of parameters. Experiments on the Merton model (unpublished) have shown that it is difficult to recover the parameters accurately for 'small' average jumps of -10% or -20%, as many parameter values result in price errors close to zero. In other words, the optimization works well and the prices can be fitted well, but the various parameters cannot be determined accurately. In any case, the parameter values used here are consistent with those reported in the literature[22]. The accuracy of the large jumps has been improved. This is consistent with the results of other studies. For example, He[23] reported a relatively accurate estimate of the mean jump size of -90%. (This also raises the question of how reasonable the parameters should be). The advantage of the theoretical model over simple interpolation is that the parameters can be interpreted. There is little benefit in using such a model if only unrealistic parameters can be used to fit option prices.

It can be seen that even when the results in terms of the objective function are good, i.e. the price errors are small, the errors in the estimated parameters can be large. This may be due to the particular parameters of Merton's jump-diffusion model described above. It also reflects the fact that both stochastic volatility and jump models can produce volatility smiles (although stochastic volatility models perform better in long periods, while jump models perform better in short periods[24] ). Therefore, the optimization procedure cannot unambiguously identify the cause of the smile. This is a problem of model identification rather than a numerical approach.

The objective function can be rewritten as a system of non-linear equations $\frac{\left\lfloor C_i^{\text{mod}el} - C_i^{market} \right\rfloor}{C_i^{market}} = 0$

Where $i \in 1, \cdots, M$ . Since the number of market prices, M, is greater than the number of parameters, the system is over-identified and can only be resolved by minimizing the residual criterion. The conditionality of the equation does not necessarily affect the value of the residuals. A poorly conditioned equation can also lead to small residuals, but in this case the parameters cannot be defined precisely. This is the same as in the present case.

The results show that the condition numbers in the model are mostly numerically acceptable with double precision (e.g. of the order of 105 or 106), although the conditions deteriorate sharply in some steps.

The fact that the condition numbers are not numerically problematic does not mean that the model is not problematic. Intuitively, for a linear regression model, the Jacobian is the data matrix. The number of conditions in this matrix may be acceptable, even if the correlation between columns is too high to draw reasonable inferences. The Matlab scenario below sets up a linear regression model with highly correlated regressors.

There are no numerical problems with the calculation of the regression coefficients, but it is best not to interpret them.

## 6. CONCLUSIONS

The authors discuss the problem of calibrating the option pricing model, where all methods are consistent in terms of cost, but the parameter estimates converge very slowly and do not converge to the jumps in the Bates model. In contrast, the parameters of the Heston model were much easier to estimate. Empirical studies testing the effectiveness of hedging should take into account the sensitivity of the results to calibration. However, optimization validity should be tested empirically rather than by hypothesis. Such a test is simple. By running a series of empirical tests, repeating the calibration with different initial values each time, the sensitivity of the results to the quality of the optimisation can be determined. The results of this study suggest that model builders in quantitative finance should be sceptical about pure numerical accuracy. Model risk remains an underappreciated aspect of quantitative finance (and one that is best not addressed by rigorous mathematical models). This result suggests that lower levels of numerical optimisation may themselves introduce bias.

## ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my professor. It was under his careful guidance that I was able to delve into the field of financial mathematics. He has provided me with unfailing guidance and care. His rigorous approach and keen academic insight have set an academic example for me and have always inspired me to keep going.

I would like to thank all the teachers in the School of Mathematical Sciences who have taught me the courses and who have given me a solid foundation in mathematics. I would like to thank all my peers and friends for their help and guidance in my academic studies.

## REFERENCES

[1]Bachelier,L. Random Character of Stock Market Prices. Theorie de la Speculation, Doctor Thesis, Annales Scientifiques Ecole Normale Sperieure III -17, 1900

[2]Markowitz，H. Harry. Portfolio Selection: Efficient Diversification of Investment. Journal of the Institute of Actuaries, 1992

[3]WF Sharpe. A Simplified Model for Portfolio Analysis. INFORMS , 9 (2) :277-293, 1963

[4]Edrington, L. H. The Hedging Performance of the New Futures Markets. Journal of Finance, 34, 157-170, 1979

[5]Modigliani and Miller. The cost of capital, corporation finance and the theory of investment. The American Economic Review, Vol. 48, p3, 1958,

[6]Brennen. Research on the Capital asset pricing model, Journal of economy, 1973

[7]Samuelson, P. A. Proof that properly anticipated prices fluctuate randomly, Industrial Management Review 6, 41 – 49, 1965

[8]Barry Schachter，J.S. Butler. The statistical properties of parameters inferred from the black-scholes formula. International Review of Financial Analysis,1996

[9]Stephen A.Ross. The valuation of options for alternative stochastic processes. Journal of Financial Economics, Volume 3, Issues 1 – 2, January – March, Pages 145-166, 1976

[10]Jim Gatheral. The Volatility Surface. Wiley, 2006.

[11]Hansjörg Albrecher, Philipp Mayer, Wim Schoutens, and Jurgen Tistaert. The Little Heston Trap. Wilmott, pages 83–92, January 2007.

[12]David S. Bates. Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options. Review of Financial Studies, 9(1):69–107, 1996.

[13]Philip J. Davis and Philip Rabinowitz. Methods of Numerical Integration. Dover, 2nd edition, 2007.

[14]Nicholas Hale and Lloyd N. Trefethen. New quadrature formulas from conformal maps. SIAM Journal of Numerical Analysis, 46(2):930–948, 2008.

[15]Lloyd N. Trefethen. Is Gauss Quadrature Better than Clenshaw–Curtis? SIAM Review, 50(1):67–87, 2008.

[16]Fiodar Kilin. Accelerating the Calibration of Stochastic Volatility Models. Centre for Practical Quantitative Finance Working Paper Series No. 6, 2007.[24]Sanjiv Ranjan Das and Rangarajan K. Sundaram. Of Smiles and Smirks: A Term Structure Perspective. The Journal of Financial and Quantitative Analysis, 34(2):211– 239, 1999.

[17]Rainer M. Storn and Kenneth V. Price. Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 11(4):341–359, 1997.

[18]Russell C. Eberhart and James Kennedy. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micromachine and Human Science, pages 39–43, Nagoya, Japan, 1995.

[19]John A. Nelder and Roger Mead. A Simplex Method for Function Minimization. Computer Journal, 7(4):308–313, 1965.

[20]W. Spendley, G.R. Hext, , and F.R. Himsworth. Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. Technometrics, 4(4):441–461,1962.

[21]Margaret H. Wright. Direct search methods: Once scorned, now respectable. In D.F. Griffiths and G.A. Watson, editors, Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, pages 191–208. Addison Wesley Longman, 1996.

[22]Kai Detlefsen and Wolfgang K. Härdle. Calibration

Risk for Exotic Options. Journal of Derivatives, 14(4):47–63, 2007.

[23]C. He, J.S. Kennedy, Thomas F. Coleman, Peter A. Forsyth, Y. Li, and K.R. Vetzal. Calibration and Hedging under Jump Diffusion. Review of Derivatives Research, 9(1):1–35, 2006.