Spring 2014

Investor expectations of the 2007-2009 Financial Crisis: Applying the Bates model to modern Stock Market events

Eric Gene Kitchens
James Madison University

Follow this and additional works at: https://commons.libjmu.edu/honors201019

Recommended Citation
https://commons.libjmu.edu/honors201019/438

This Thesis is brought to you for free and open access by the Honors College at JMU Scholarly Commons. It has been accepted for inclusion in Senior Honors Projects, 2010-current by an authorized administrator of JMU Scholarly Commons. For more information, please contact dc_admin@jmu.edu.

A Project Presented to

the Faculty of the Undergraduate

College of Business

James Madison University

in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science

by Eric Gene Kitchens

May 2014

Accepted by the faculty of the Department of Finance, James Madison University, in partial fulfillment of the requirements for the Degree of Bachelor of Science.

FACULTY COMMITTEE:

Project Advisor: Jason Fink, Ph.D.,
Eminent Professor, Finance

Reader: Kristin Fink, Ph.D.,
Professor, Finance

Reader: Elias Semaan, Ph.D.,
Associate Professor, Finance

HONORS PROGRAM APPROVAL:

Barry Falk, Ph.D.,
Director, Honors Program
Table of Contents

Acknowledgements .................................................. 3
Abstract ................................................................ 4
Introduction ............................................................ 5
Option Pricing Models ................................................. 11
Estimation ............................................................... 21
Results .................................................................. 25
Discussion .............................................................. 30
Conclusion .............................................................. 34
Appendix A: Figures .................................................... 35
Appendix B: Tables ..................................................... 53
Appendix C: Matlab Code ............................................ 55
References .............................................................. 65
Acknowledgements

I would like to thank Dr. Jason Fink, without whom I would not have been able to explore my potential and academic curiosities. His patience, tutelage, guidance, and influence have been invaluable to me and to this project. My experience at James Madison would not have been the same without you.

I would also like to thank my readers, Dr. Kristin Fink and Dr. Elias Semaan, for their feedback, support, and enthusiasm. Thank you each for being excited and engaged in the spirit and content of this project from the proposal to the final draft.

I would like to thank Dr. Barry Falk and the Honors Program, which provided substantial aid for purchasing the data necessary for my research with an Honors Program Small Grant award. Thank you to Dr. Pamela Drake and the Finance Department for also contributing significant funds for the data.

Finally, I would like to thank my family, friends, and coworkers for their continuous understanding and flexibility. To my wife, Katie, thank you for allowing me to vent, think aloud, rant excitedly, and stay on task. I could not have done this without you.
Abstract

Following the financial crisis of 2007-2009, many analysts argued that market insiders should have anticipated the crisis. We update and apply the Bates (1991) method to see if investors anticipated a stock market crash during this period. S&P 500 options prices are used to estimate daily implicit parameters from the constant volatility jump diffusion (CVJD) model used by Bates, a stochastic volatility no-jump (SV) model, and a stochastic volatility jump diffusion (SVJD) model, and we analyze the implicit investor expectations of large negative price jumps in the stock market. Rather than finding any predictions of the crisis, we find evidence of efficient markets as the parameters implied by S&P 500 options move in step with the market as a whole.
Introduction

Many economists agree that the financial crisis of 2007-2009 was the worst since the great depression.\(^1\) In hindsight, it seems that the bursting of the U.S. housing bubble and heavily-leveraged trading in mortgage-backed securities (MBS) led to the large fall in the markets (see Taylor, 2009). The U.S. stock market crash of 1987 had been similarly subject to backward-looking explanations, but in 1991 Bates found evidence that the crash was *anticipated* by investors. For his analysis, Bates developed a constant-volatility jump-diffusion (CVJD) option pricing model and applied it to historical prices of options on S&P 500 futures, with which he estimated the jump parameters implied by out-of-the-money options (Bates, 1991). Bates used time series plots of the daily implicit values of these parameters to explore market participants’ expectations for the frequency, size, and direction (positive or negative) of discrete jumps in stock prices in the months surrounding and including October 1987 (Bates, 1991). He found that crash expectations spiked two months before the crash, effectively predicting the event.

Would we find the same with the financial crisis, considering we now believe that the U.S. housing bubble burst in 2006 (see Martin, 2011)? We answer this question with methodology that uses an improved option pricing model to apply Bates’ methodology to the financial crisis. The key improvement is the inclusion of stochastic volatility. In 1996, Bates updated his CVJD model to allow for stochastic volatility as well as jump diffusion parameters (the SVJD model). Bakshi, Cao, and Chen (1997) compared models that incorporated stochastic volatility, jump diffusion, or both when pricing S&P 500 options, and found that stochastic volatility was the most important factor for the internal consistency of the model and for applications involving hedging. Fink and Fink (2006) found that the Bates’ (1996) SVJD model

outperformed the classic Black-Scholes (B-S) model (Black and Scholes 1973), as well as a CVJD model by Merton (1976) and a no-jump stochastic volatility (SV) model by Heston (1993), in terms of consistency and accuracy when performing Monte Carlo simulations to price Google options. Therefore, the SVJD modeling system should offer a significant increase in accuracy and retain the implicit jump parameters necessary to analyze investor expectations. To allow comparison, we also estimate the parameters of the B-S, CVJD, and SV models in a similar way.

**Key Dates and Trends**

In order to thoroughly investigate anticipations of market movements, an overview of market trends and large price movements is necessary. The S&P 500 index’s broad range of equities make it a good proxy for the U.S. stock market as a whole, and as Bates (1991) explains, options on this index are a good indicator of investor expectations. For our analysis, we will examine the index from 2006 to 2010.

**General Trends.** At the beginning of 2006, the S&P was stable and increasing (see Figure 1, Appendix A). With few exceptions, this trend continued until July 2007, when major price swings became larger and more frequent. Afterwards, the index followed up-and-down cycles through its October 2007 peak and continued to cycle until August 2008. The most striking period of market decline during the financial crisis occurred between August 2008 and March 2009, when we see the most significant discrete index price movements. Although scholars disagree on the causes of the financial crisis, this is still the relevant period of reference for determining whether investors actually *predicted* any large negative jumps in the overall stock market. We will casually refer to the notable decline in the market over this period as the

---

2 All figures may be found in Appendix A.
market “crash.” When the index began to rise from its lowest point (in March 2009), the sizes and frequency of price spikes remain heightened relative to 2006 levels, and the market trended upward, despite some up-and-down cycles with short periods, including the days surrounding the May 2010 Flash Crash.

**Key Dates.** We will analyze the behavior of the various models’ parameters on and before the trading days with the largest percentage changes in the index price. Therefore, we begin with a recap of major financial events, which will allow us to determine whether implicit jump expectations were anticipatory or reactionary.\(^3\) For convenience, Table 1, Appendix B, lists some of the largest S&P price movements during (A) the beginnings of the financial crisis, (B) the height of the financial crisis, and (C) the period surrounding the Flash Crash.\(^4\)

Perhaps the first large index drop related to the crisis came on February 27, 2007 when Freddie Mac publically distanced itself from the riskiest MBS and the S&P fell 3%.\(^5\) After this, the market resumed a slow and steady incline until June 7, 2007, when Bear Stearns wrote a letter to its investors announcing a suspension of redemptions from its primary MBS funds.\(^6\) This announcement coincided with a modest 2% drop in the index, and when Bear Stearns filed for bankruptcy protection for those funds on July 31, 2007, the market had already been dropping over the course of the previous week and only fell 1% that day. All of these events occurred

---

\(^3\) The Federal Reserve Bank of St. Louis compiled a comprehensive timeline of events related to the financial crisis and links to relevant documents. This resource was heavily referenced for this section of the paper, and is available here: http://timeline.stlouisfed.org/index.cfm?p=timeline

\(^4\) All tables may be found in Appendix B.


during an overall upward trend in the market.\textsuperscript{7} On March 14, 2008, when Bear Stearns was bailed out by the New York Federal Reserve with assistance from J.P. Morgan, the index dropped another 2%, but had already been declining steadily for months.\textsuperscript{8}

The most volatile time during the financial crisis was in mid-September to early December 2008, a period that includes the largest percentage increases and decreases in the S&P during the observed window (see Table 1, panel B) and perhaps the most likely timeframe for crash expectations. On September 15\textsuperscript{th}, Lehman Brothers filed for bankruptcy and by September 17\textsuperscript{th} the Securities and Exchange Commission banned short selling of the stocks of financial companies as expectations of other bank failures were high.\textsuperscript{9,10} The index dropped 5% on each of those days, but recovered 4% on each of the subsequent days. On September 29\textsuperscript{th}, when the U.S. House of Representatives rejected a bailout bill that had passed in the Senate, the index fell 9% with a 5% recovery the following day. In the midst of the worst week ever observed for the Dow Jones Industrial Average, the S&P itself fell a whopping 8% on October 9\textsuperscript{th}.\textsuperscript{11,12} On the eve of the press conference to unveil the $700 billion bailout fund within the Troubled Asset Relief Program (TARP), the S&P shot up 12% only to fall 9% two days later when the Federal Reserve Chairman gave a speech implying that TARP may not do enough to strengthen the U.S. financial

\textsuperscript{7} http://www.nysb.uscourts.gov/opinions/brl/158969_24_opinion.pdf
\textsuperscript{9} http://www.sec.gov/rules/other/2008/34-58572.pdf
\textsuperscript{10} http://www.sec.gov/Archives/edgar/data/806085/000110465908059632/a08-22764_4ex99d1.htm
system. On October 28th, when the U.S. Treasury invested $125 billion from its Capital Purchase Program (a subset of TARP) to stabilize the nine largest financial institutions, the index rose 11%. Then, on November 13th the Treasury announced that it would inject TARP funds directly into large banks rather than purchasing illiquid MBS—effectively bolstering financial institutions rather than attempting to stave off foreclosures—the S&P spiked 7%. And when Goldman Sachs Group posted a $2 billion loss for the quarter ending in November 2008, Goldman’s stock dropped 18% while many other financial institutions’ stocks dropped and the S&P 500 index dropped 9%.

The Flash Crash of May 2010 provides an opportunity to examine how the models reflect investor reactions to large market movements, since this crash could not have been anticipated. If we compare how the models' parameters behave in the days surrounding the Flash Crash to the parameters' behavior during the financial crisis, we can see whether anticipation or reaction is reflected in the data. From March 2009 to April 2010, the S&P 500 index followed a slow and relatively steady march upward and over $1,200 mark for the first time since the Lehman Brothers bankruptcy. But, on May 6, 2010, at approximately 2:45 PM eastern time, several blue chip stocks crashed within five minutes, likely due to a glitch in a computerized trading algorithm. Although the stocks mostly rebounded and normalcy resumed, the S&P 500 index

---

still finished down 3% and down another 2% the following day (See Table 1, panel C). On Monday, May 10, the index did rebound 4%, almost returning to its May 5th level. Ten days later, the index dropped 4% when the Senate announced a Wall Street reform bill that some analysts estimated would cause an annual profit loss of up to 20% for the largest investment banks. But, on May 21st the index returned to a pattern of slow and steady increase that lasted until the end of 2010.

Option Pricing Models

To better understand the SVJD model, we will review how the model developed from previous models. The first option pricing model of this kind was the famous Black-Scholes option pricing formula from their 1973 paper. Merton (1973) co-developed the B-S model, but in a 1976 paper, he provided theoretical justification for his CVJD model and developed another option pricing formula.\(^\text{20}\) This is the same formula and theory that Bates applied in his 1991 paper on the stock market crash of October 1987. Then, Heston (1993) developed an option pricing formula for an SV model supported by research which showed that spot volatility is not constant over time, as assumed in the B-S model. And in 1996, Bates published an option pricing formula for an SVJD model including both discontinuous jumps and stochastic volatility.

**Black-Scholes (B-S) Constant Volatility Model**

The Black-Scholes model is the most famous option pricing model in the literature and is the basis for the other three models that we will apply to the data.\(^\text{21}\) In defining their European option pricing model, Black and Scholes assumed the following:

a.) The risk-free interest rate is known and is constant, and market participants can borrow or lend any amount at this rate.

b.) Stocks or other assets do not pay dividends.

c.) There are no transaction costs or volume restrictions in buying or selling (or short-selling) the stock or the option, nor while borrowing or lending at the risk-free rate,


and taxes are the same regardless of whether income is from capital gains or dividends. This is known as the frictionless market assumption.

d.) Asset prices follow a Geometric Brownian Motion (GBM) process, and therefore future asset prices are distributed log-normally.

Questions of the feasibility of assumption (d) were the driving force leading to the CVJD, SV, and SVJD models. As Merton (1976) catalogued, research has shown that the Black-Scholes modeling system is still valid even if (a) interest rates are not constant over time, (b) assets pay dividends, and (c) markets are not completely frictionless. The GBM assumption asserts that the price of the underlying asset changes continuously over time, and that the variance of the lognormal distribution is constant over time. Under this assumption, a European call option can be calculated as:

\[ C = N(d_1)S e^{-\delta \tau} - N(d_2)X e^{-r \tau}, \]

\[ d_1 = \frac{\ln \left( \frac{S}{X} \right) + \left( r - \delta + \frac{\sigma^2}{2} \right) \tau}{\sqrt{\sigma^2 \tau}}, \]

\[ d_2 = d_1 - \sqrt{\sigma^2 \tau}, \]

where \( C \) is the price of the put, \( S \) is the price of the underlying asset, \( X \) is the strike price or exercise price of the option, \( r \) is the continuously compounded (annualized) risk-free rate, \( \delta \) is the continuously compounded (annualized) dividend yield of the underlying asset, \( \tau \) is the time

---

to expiry, \( \sigma \) is the constant volatility of returns on the underlying asset, and \( N(\cdot) \) is the cumulative standard normal distribution.\(^{23}\)

Similarly, the value of a European put option \( (P) \) under the Black-Scholes model is:\(^{24}\)

\[
P = N(-d_2)Xe^{-r\tau} - N(-d_4)Se^{-\delta \tau}.
\]

Alternatively, a put may be valued according to put-call parity, an equation that holds regardless of the model used to calculate \( C \) and \( P \) (see McDonald, Cassano, and Fahlenbrach, 2006):

\[
P = C - Se^{-\delta \tau} + Xe^{-r\tau}.
\] \hfill (2)

Under the Black-Scholes framework, the only unknown value when calculating option prices is the constant volatility parameter \( \sigma \). By plugging in all the known characteristics of an option—\( S, X, r, \delta, \tau \)—and the market price of the option, one can determine the “implied” volatility by solving the appropriate above equation for \( \sigma \). This implied volatility is the market consensus of the true volatility as determined by option prices on the open market, given the assumptions of the B-S model. As Bakshi, Cao, and Chen (1997) point out, implied parameters—found by fitting option pricing formulas to market values for the options—are necessarily forward-looking, since options are bets or insurance on future price movements. Hence, implicit parameters show investors’ expectations for the relevant asset, and since the collection of assets in the S&P 500 index is broad, implicit parameters from options on this index show expectations for the market as a whole.

**Constant Volatility with Jump Diffusion (CVJD) Model**

---

\(^{23}\) Dividend yields were excluded by Black and Scholes (1973) but were included by Merton (1973).

In his 1976 paper, Merton introduced the CVJD model, which allows for asset prices to change both continuously and discontinuously, via a stochastic jump process. Merton conceded that the Black-Scholes analysis is not invalidated by the fact that price changes cannot be wholly continuous and that the Black-Scholes equation holds if the sample path of the asset price approaches a continuous sample path as the time between price changes approaches zero. However, Merton counters that the Black-Scholes argument is invalid if the sample path of the asset does not asymptotically approach a continuous process, which is generally found in the paths of asset prices (see Akgiray and Booth, 1986). To correct for this, Merton adds a stochastic jump diffusion process to the Black-Scholes model, where discontinuous price jumps arrive via a Poisson process and the GBM assumption holds in the absence of a price jump. The resulting call option pricing formula is:

\[
C = \sum_{n=0}^{\infty} \frac{e^{-\lambda(1+\mu)\tau}}{n!} \left( \lambda (1 + \mu) \tau \right)^n f_n,
\]

\[
f_n = N(d_1)S e^{-\delta \tau} - N(d_2)X e^{-r \tau},
\]

\[
d_1 = \frac{\ln \left( \frac{S}{X} \right) + \left( r - \delta - \lambda \mu_j + \sigma_j^2 \right) \tau + n \ln (1 + \mu_j) + \frac{n \sigma_j^2}{2}}{\sqrt{\sigma^2 \tau + n \sigma_j^2}},
\]

\[
d_2 = d_1 - \sqrt{\sigma^2 \tau + n \sigma_j^2}, \tag{3}
\]

where \( \sigma \) is the constant volatility of the continuous portion of the process, \( \lambda \) is the mean number of discontinuous jumps in asset prices per year (governed by a Poisson process), \( \mu_j \) is the mean jump size (as a percentage of the asset price) given that a jump occurs, \( \sigma_j \) is the standard deviation of jump sizes, and \( S, X, r, \delta, \tau, \) and \( N(\cdot) \) are defined as above in the Black-Scholes case. Again, the price of a put option may be calculated by put-call parity as in equation (2).
Here, $\sigma$ is volatility conditional on no jumps occurring. The variance (volatility squared) given that a jump occurs can be calculated as:

\[
\text{jump variance} = \lambda \left[ \left( \ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2 \right)^2 + \sigma_j^2 \right].
\]

The unconditional spot variance is the sum of the variance conditional on no jumps occurring and the variance given that a jump occurs. The square root of this unconditional variance is the spot volatility, given by:

\[
\text{spot volatility} = \sqrt{\sigma^2 + \lambda \left[ \left( \ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2 \right)^2 + \sigma_j^2 \right]}.
\]

This model has four unknowns, the continuous volatility $\sigma$, the unconditional mean jump size (as a percentage of the spot price) $\mu_j$, the annual frequency of jumps $\lambda$, and standard deviation of jump size $\sigma_j$. Bates (1991) used this model’s implied parameters from S&P 500 European option data from 1985-1987 to find investors’ jump risk expectations prior to and immediately after the October 19-20 stock market crash in 1987. A key insight by him was to plot the product $\lambda \mu_j$ daily to show a proxy for the expected annual change in price due to discontinuous jumps. This time series plot showed that two months before the crash occurred, expectations of severe negative price jumps were implicit in options prices. While the $\lambda \mu_j$ plot is instructive, it does not have a mathematically intuitive meaning. This paper will instead plot the implicit mean annual change in asset prices due to discontinuous jumps, calculated as:

\[
\text{jump expectations} = (1 + \mu_j)^\lambda - 1.
\]

Notice that both the CVJD and B-S model assume that the continuous volatility parameter $\sigma$ is constant. Under this assumption, the implied $\sigma$ for options on a single asset should be constant across different strike prices. However, this is not observed in the data, and
instead volatility “smiles”, “smirks”, or “frowns” are often observed (see Fink and Fink, 2006), indicating that volatility may also change over time.

**Stochastic Volatility (SV) Model**

Heston (1993) introduced an option pricing model where asset prices are assumed to follow a continuous sample path as in the B-S framework, but volatility is modeled using a stochastic mean-reversion process and the innovations of asset price and volatility are correlated. Using the parameter notation used by Fink and Fink (2006) and the equation structure used by Epps (2000), the SV model prices options as follows:

\[
P = X e^{-r \tau} \tilde{F}(X) - S e^{-\delta \tau} \tilde{G}(X),
\]

where \( \tilde{F}(X) \) and \( \tilde{G}(X) \) are defined by \( J \in \{F, G\} \) as follows:

\[
\tilde{f}(X) = \frac{1}{2} - \lim_{\varepsilon \to \infty} \frac{1}{\varepsilon} \int_{-\varepsilon}^{\varepsilon} \frac{\Psi_J(\zeta; s, \nu, \tau)}{2\pi i \zeta} e^{-i \zeta \ln X} \cdot d\zeta,
\]

\[
s = \ln S,
\]

\[
\Psi_J(\zeta; s, \nu, \tau) = \exp \left[ g_J(\tau; \zeta) + h_J(\tau; \zeta) \nu + i \zeta s \right],
\]

\[
g_J(\tau; \zeta) = i \zeta (r - \delta) \tau + \frac{\theta \nu}{\sigma^2} \left[ (b_J - i \zeta \rho \sigma_v + D_J) \tau - 2 \ln \left( \frac{1 - Q_J e^{\tau D_J}}{1 - Q_J} \right) \right],
\]

\[
h_J(\tau; \zeta) = \frac{i \zeta \rho \sigma_v - b_J - D_J}{\sigma_v^2} \cdot \frac{e^{\tau D_J} - 1}{1 - Q_J e^{\tau D_J}},
\]

\[
Q_J = \frac{B_J - D_J}{B_J + D_J},
\]

\[
D_J^2 = B_J^2 - 2 A_J \sigma_v^2,
\]

---

25 Call option prices may be calculated using put-call parity.
\[ A_j \equiv i\sigma_j - \frac{\sigma_j^2}{2}, \]
\[ B_j \equiv i\rho\sigma_j - b_j, \]
\[ a_j \equiv \begin{cases} -\frac{1}{2}, & J = F \\ +\frac{1}{2}, & J = G \end{cases}, \]
\[ b_j \equiv \begin{cases} \kappa_\nu, & J = F \\ \kappa_\nu - \rho\sigma_\nu, & J = G \end{cases}, \] (5)

where

\[ \frac{dS}{S} = (r - \delta)dt + \sqrt{\nu_t}dW_\nu, \]
\[ d\nu_t = [\theta_\nu - \kappa_\nu\nu_t]dt + \sigma_\nu\nu_t dW_\nu, \]
\[ \rho = Corr(dW_\nu, dW_\nu). \]

Here, \( \nu_t \) is the instantaneous variance of asset price (volatility squared); \( \theta_\nu \) and \( \kappa_\nu \) are parameters governing the mean reversion process of the instantaneous variance; \( \sigma_\nu \) is the volatility of volatility; \( dW_\nu \) and \( dW_\nu \) are Weiner processes associated with price and volatility, respectively; and the inputs \( S, X, r, \delta, \) and \( \tau \) are defined as above. Since, the SV model has no jump component, the parameter \( \nu_t \) is unconditional. Therefore, the spot variance is simply \( \sqrt{\nu_t} \), and the long-run volatility is \( \frac{\theta_\nu}{\sqrt{\kappa_\nu}} \).

**Stochastic Volatility with Jump Diffusion (SVJD) Model**

In his 1996 paper, Bates referenced previous research showing that asset price variance is not constant—a theoretical validation for the SV model—and that asset price sample paths sometimes involve discontinuous jumps—a theoretical validation for the jump-diffusion
model—and combined the two into the SVJD model. Like the CVJD model, the SVJD model has the necessary parameters to show whether investors expected a market crash at any time surrounding or during the financial crisis via time series plots of equation (4); but the SVJD model also allows for implied stochastic volatility and implied correlation between the innovations in asset price and volatility. Again using the Fink and Fink (2006) parameter definitions and the Epps (2000) formula structure, the SVJD option pricing formula is: 

\[ P = Xe^{-\tau r} \tilde{F}(X) - Se^{-\delta \tau} \tilde{G}(X), \]

where \( \tilde{F}(X) \) and \( \tilde{G}(X) \) are defined by \( J \in \{F, G\} \) as follows:

\[ f(X) = \frac{1}{2} - \lim_{\epsilon \to \infty} \int_{-\epsilon}^{\epsilon} \frac{\Psi_j(\varsigma; s, \nu, \tau)}{2\pi i \varsigma} e^{-i\varsigma \ln X} d\varsigma, \]

\[ s = \ln S, \]

\[ \Psi_j(\varsigma; s, \nu, \tau) = \exp \left[ g_j(\tau; \varsigma) + h_j(\tau; \varsigma) \nu + i\varsigma s + \tau \gamma_j \right], \]

\[ g_j(\tau; \varsigma) = i\varsigma (r - \delta - \lambda j) + \frac{\theta_v}{\sigma_v^2} \left( (b_j - i\varsigma \rho \sigma_v + D_j) \tau - 2 \ln \left( \frac{1 - Q_j e^{\tau D_j}}{1 - Q_j} \right) \right), \]

\[ h_j(\tau; \varsigma) = \frac{i\varsigma \rho \sigma_v - b_j - D_j}{\sigma_v^2} \cdot \frac{e^{\tau D_j} - 1}{1 - Q_j e^{\tau D_j}}, \]

\[ Q_j = \frac{B_j - D_j}{B_j + D_j}, \]

\[ D_j^2 = B_j^2 - 2A_j \sigma_v^2, \]

\[ A_j = i\varsigma a_j - \frac{\varsigma^2}{2}, \]

\[ B_j = i\varsigma \rho \sigma_v - b_j, \]

---

26 Call option prices may be calculated using put-call parity.
where the inputs and have their usual definitions; the jump-diffusion parameters — and — are defined as in the CVJD process; and the stochastic volatility parameters — and — are defined as in the SV process. In this model, is the instantaneous spot variance conditional on no jumps, similar to in the CVJD model. Therefore, the volatility due to the discontinuous jump process and the unconditional spot volatility are similar to the CVJD case:

\[
\frac{dS}{S} = [r - \delta - \lambda \mu_j]dt + \sqrt{v_\tau} dW_\tau + J_\tau dq_\tau,
\]

\[
v_\tau = [\theta_\nu - \kappa_\nu v_\tau]dt + \sigma_\nu v_\tau dW_\nu,
\]

\[
\rho = \text{Corr}(dW_\nu, dW_\nu),
\]

\[
\ln(1 + J_\tau) \sim N \left( \ln(1 + \mu_j) - \frac{\sigma_j^2}{2}, \sigma_j^2 \right).
\]

The inputs , , , , and have their usual definitions; the jump-diffusion parameters — and — and — are defined as in the CVJD process; and the stochastic volatility parameters — , , and — are defined as in the SV process. In this model, is the instantaneous spot variance conditional on no jumps, similar to in the CVJD model. Therefore, the volatility due to the discontinuous jump process and the unconditional spot volatility are similar to the CVJD case:

\[
\text{jump variance} = \lambda \left( \ln(1 + \mu_j)^2 - \frac{1}{2} \sigma_j^2 \right) + \sigma_j^2,
\]

\[
\text{spot volatility} = \sqrt{v_\tau + \lambda \left( \ln(1 + \mu_j)^2 - \frac{1}{2} \sigma_j^2 \right) + \sigma_j^2}.
\]
Jump expectations for the SVJD model may be calculated using equation (4), just as with the CVJD model.

\[
\text{long-run volatility} = \sqrt{\frac{\theta_v}{\kappa_v} + \lambda \left[ (\ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2)^2 + \sigma_j^2 \right]}. 
\]
Estimation

Data

Option Data. Option data on the S&P 500 index from January 2006 through December 2010 was obtained from the Chicago Board Options Exchange’s historical data retailer, Market Data Express. This data lists daily information on all European options available for trading. For each option the following information was listed: the trade date, that day’s S&P closing price, the exercise date of the option, the strike price, whether the option was a call or a put, the open interest in the option as of that date, the trade volume of the option on that date, and the last bid and last ask prices on the option for that date.

Certain constraints were placed on the data so that parameter optimization was only applied to the most suitable data. Only options with more than five days and fewer than six months to expiry were considered, since options expiring within five days convey little about future market sentiment and options expiring more than 6 months out fail to convey investor’s short term expectations. Any option that was available but was not traded at all on a certain day (volume equal to zero) was removed from that day’s data, because such options would not be relevant to that day’s implied parameters. All options with an open interest level less than 50 on a given day were excluded as a way of removing thinly traded options. Of all options in the data package, the median open interest was 8,298 and the cutoff open interest level of 50 was approximately the 10\textsuperscript{th} percentile.

For optimization, the market price of each option was calculated as the midpoint between the last bid and last ask prices—the raw bid-ask spread. Options with a percentage spread greater

\footnote{The product is called Optsum and was purchased from www.marketdataexpress.com via a link on www.cboe.com.}
than 50% were excluded in order to avoid misrepresentation of market trading value for such options. The final data set consists of 262,962 options across 1,259 trading days (208.8658 per day), with 108,658 calls (86.3050 per day) and 154,304 puts (122.5608 per day).

**Interest Rate and Dividend Yield Data.** Risk-free rates were estimated using 3-month U.S. Treasury Bill yields as listed in the Fama-French Liquidity Factors dataset housed in the Wharton Research Data Services (WRDS) database.\(^2^9\) One risk-free rate was used for all options within a given trading month, since Bates (1996) and Scott (1993) found this simplification has little impact on analysis of relatively short-dated options. The Fama-French website lists the daily effective rate (based on 251 trading days per year), \(\frac{i^{(251)}}{251}\). This rate was converted to a continuously compounded rate, \(r = \ln \left(1 + \frac{i^{(251)}}{251}\right)^{251}\).

Dividend yields were calculated using data available on Dr. Robert Shiller’s website.\(^3^0\) Since earnings and dividends of the S&P 500 data were provided monthly, within each trading month the same dividend yield was assumed. A nominal monthly dividend yield was calculated as \(i^{(12)} = \frac{\text{monthly dividends}}{\text{monthly index price}}\). Then, a continuously compounded dividend yield was calculated as \(\delta = \ln \left(1 + \frac{i^{(12)}}{12}\right)^{12}\).

**Parameter Optimization**

Once the data was constrained as above and the appropriate risk-free rates and dividend yield rates were included in each month’s data, separate data files were created for each trading

\(^{28}\) *percentage spread* = \(\frac{\text{last ask price} - \text{last bid price}}{\text{last trade price}}\)

\(^{29}\) http://www.wrdsweb.wharton.upenn.edu/wrds/ds/famafrench/factors_d.cfm

day’s options. As Bates (1991) stated, estimating new parameters for every trading day may be appear to violate the option pricing models’ assumption that the parameters either are constant or change slowly over time, but this daily estimation method provides vital insight about how market sentiments are changing over time. We must keep this potential inconsistency in mind when we interpret the estimated daily parameters.

For all four models discussed above, implicit daily parameters were estimated via non-linear least squares using a trust region reflexive algorithm. Every option within a day’s data listed the index price $S$, whether the option was a call or a put, the risk-free rate $r$, the dividend yield $\delta$, the strike price $X$, and the time to expiry $\tau$. This input data was fit to the options’ market prices by non-linear least squares optimization of the parameters of the particular option pricing model (B-S, CVJD, SV, or SVJD). For each parameter, lower and upper bounds were stipulated to speed computation and to maintain consistency with the meanings of each parameter (see Table 2). Additionally, an initial guess at each parameter’s value is supplied. To reconcile the potential inconsistencies of re-estimating parameters daily, each day’s initial-guess vector was set equal to the previous day’s estimated parameters.

For each model, the least squares algorithm was applied to each trading day’s data individually, simultaneously optimizing the parameters across all option data for that day. The estimation process resulted in daily vectors of parameters that minimize the entire trading day’s

\[31\] The least squares fit was completed using the LSQCURVEFIT function in MATLAB. See Appendix C for the code.
\[32\] For the CVJD model, the infinite sum of equation (3) was truncated at $n=k$ if the absolute difference in option price changed by less than 0.000001 when the summand for $n=k+1$ was added. Similarly, the bounds of the integrands for the SV and SVJD model, equations (5) and (6), were estimated by increasing $c$ by increments of 1 until the absolute difference in $F$ or $G$ going from $c=k$ to $c=k+1$ was less than 0.000001, with independent truncation points for $F$ and $G$. Each integral calculation was numerically computed using Gaussian quadrature via Matlab’s “integral” function (see Appendix C for the code).
sum of squared errors between expected option prices (according to the particular model) and the observed trading prices of the options.
Results

Table 3 displays the pricing error measurements for each model’s estimated parameters. As expected based on previous research, the CVJD, SV, and SVJD models all provided average absolute pricing error reductions relative to B-S. As expected, the SVJD model outperformed all other models in every measure of performance, and the SV and CVJD models consistently outperformed the B-S model.

Black-Scholes Implied Parameters

Figure 2 shows a time series plot of implied volatility $\sigma$ from the Black-Scholes model. In 2006, implied volatility is fairly steady and low relative to the historical average of 19.5%. In mid-2007, investors began to expect more volatility in the market and a sustained period of heightened implied volatility begins just before Lehman Brothers’ collapse and $\sigma$ does not return to the 20% mark until the end of 2009. Since $\sigma$ is the only unknown parameter in the Black-Scholes framework, upon which each of the other models is built, this parameter provides a benchmark for evaluating the market reactions of parameters in the other models. Figures 1 and 2 display the typical inverse relationship between asset price and volatility, which is a key component of the SV and SVJD models.

CVJD Implicit Parameters

Figure 3A-3F contains the time series plots of the estimated CVJD parameters: (A) the implicit annual continuous volatility (conditional on no price jumps occurring) $\sigma$, (B) the implicit standard deviation of jump sizes $\sigma_j$, and (C) the implicit annual jump volatility.

---

33 This format for expressing measurements of error is borrowed from Fink and Fink (2006).
34 Based on the daily adjusted close of the VIX from 1990 to 2005. Source: http://finance.yahoo.com/q/hp?s=%5EVIX&a=00&b=2&c=1990&d=00&e=2&f=2006&g=d
and the deviation of it from Black-Scholes implied volatility—the dashed line, (E) the mean number of discontinuous price jumps per year $\lambda$, and (F) the mean jump size as a percentage of the asset price $\mu_j$.\(^{35}\) Although the continuous volatility $\sigma$ is considered constant, the spot volatility depends both on $\sigma$ and on jump volatility, which in turn depends on $\lambda$, $\mu_j$, and $\sigma_j$. Therefore, heightened jump expectations would cause heightened spot volatility. Despite this, the spot volatility for CVJD rarely differs significantly from B-S implied volatility (see Figure 3, panel D). Figure 3, panel C shows that $\sigma$ largely follows the pattern of B-S implied volatility (Figure 2) before and after the worst of the financial crisis in September and October 2008, except that the CVJD $\sigma$ has near-zero values during this worst period. The overall implicit spot volatility for CVJD still continues to mirror B-S implied volatility because of the inclusion of jump variance.

Figure 4 shows the time series plot of CVJD implicit jump expectations, calculated using equation (4).\(^{36}\) In general, these expectations seem to move in-step with price movements of the index but with inverse direction. The only sustained period of negative jump expectations of a size found by Bates’ (1991) analysis is during the September–October 2008 period, with nothing more than spikes during less tumultuous periods. This indicates that options buyers’ expectations continued to move along with the index even during the financial crisis. We analyze implicit jump expectations in more detail below.

**SV Implied Parameters**

---

\(^{35}\) This format of plotting the implicit parameters is borrowed from Bates (1991).

\(^{36}\) The correlation between the implicit values of this version of jump expectations and the version that Bates (1991) used is 0.882994 for the CVJD model.
Figure 5A-5D contains the time series plots of the estimated SV parameters: (A) the implicit spot volatility $\sqrt{v_t}$, (B) the implicit long-run volatility $\sqrt{\frac{\theta_v}{\kappa_v}}$, (C) the implicit volatility of volatility $\sigma_v$, and (D) the implicit correlation between innovations in asset prices and volatility $\rho$. The time series plot of $\sqrt{v_t}$ largely follows Black-Scholes implied volatility ($\sigma_{BS}$), with deviations occurring because $\sqrt{v_t}$ has spikes much larger than Black-Scholes during the most volatile time periods, especially in the fall of 2008 and during May 2010. Because $\sqrt{v_t}$ follows $\sigma_{BS}$ so closely, it appears to have the same inverse relationship to asset prices as $\sigma_{BS}$, and we see this reflected in the parameter $\rho$. The plot of $\rho$ in Figure 5D is noisy and therefore patterns are difficult to detect, but the centered-moving average plot in Figure 6 is illustrative. We see that this implicit correlation does not move directly nor inversely with the index, but the correlation is always negative and $\rho$ becomes and remains very close to negative one throughout the financial crisis.

**SVJD Implicit Parameters**

Figure 7A-7I contains the time series plots for the estimated SVJD parameters: (A) the implicit continuous volatility $\sqrt{v_t}$; (B) the implicit long-run continuous volatility $\sqrt{\frac{\theta_v}{\kappa_v}}$; (C) the implicit volatility of volatility $\sigma_v$; (D) the implicit standard deviation of jump size $\sigma_J$; (E) the implicit jump volatility $\sqrt{\lambda \left[\ln(1 + \mu_J) - \frac{1}{2} \sigma_J^2\right]^2 + \sigma_J^2}$; (F) the implicit unconditional spot volatility $\sqrt{v_t + \lambda \left[\ln(1 + \mu_J) - \frac{1}{2} \sigma_J^2\right]^2 + \sigma_J^2}$ and the deviation from Black-Scholes implied volatility in the dashed line; (G) the implicit correlation of innovations in asset prices and continuous volatility $\rho$; (H) the implicit annual mean number of discrete jumps $\lambda$; and (I) the implicit mean jump size as a percentage of the asset price $\mu_J$. The plot of $\sqrt{v_t}$ (Figure 7A)
matches fairly well with the plot of $\sigma_{BS}$ (Figure 2), and therefore displays an inverse relationship between the index price and the continuous volatility. As with the SV model, this information is reflected in the parameter $\rho$. The main difference between the plots of $\sqrt{v_t}$ and $\sigma_{BS}$ is that the upward and downward spikes are much larger for $\sqrt{v_t}$. This is probably due to the fluctuation allowed by the volatility of volatility $\sigma_v$, which also has a plot that follows the pattern of $\sigma_{BS}$, indicating that when the continuous volatility increases in the SVJD model, the range of likely movements for that volatility also increases, leading to the larger spikes that are shown in Figure 7A. The time series plot of jump volatility (Figure 7E) does not seem to have a direct or inverse relationship with $\sigma_{BS}$, and the implicit jump volatility even goes nearly to zero during the height of the financial crisis. While the spot volatility graph (Figure 7F) follows $\sigma_{BS}$ fairly well leading up to and coming out of the financial crisis, it deviates heavily during the majority of the crisis and repeats this deviation after the Flash Crash. By investigating Figure 7D, we can attribute these deviations to the increased $\sigma_j$ values throughout the crisis. The plot of the correlation parameter $\rho$ (Figure 7G) is noisy, as in the SV case, but the centered moving average (Figure 8) gives a clearer picture. Unlike the $\rho$ parameter in the SV model, the correlation parameter here is sometimes strongly positive, but is consistently near negative one before, during, and after the financial crisis.

Figure 9 plots the SVJD implicit jump expectations $(1 + \mu_j)^{\lambda} - 1$, which differ greatly from the CVJD case (Figure 4). As expected from the time series plot of implicit mean jump size (Figure 7H), the jump expectations plot is not always negative, and even the negative values

$^{37}$ The correlation between the implicit values of this version of jump expectations and the version that Bates (1991) used is 0.929855 for the SVJD model.
are muted compared to the CVJD model. We also see a lag in the response to the financial crisis, since annual price jumps do not reach values below negative 20% until mid-to-late October 2008. Then, jump expectations return to positive values as early as mid-December 2008. We examine the jump expectations for both the CVJD and SVJD models in more detail below.
Discussion

The most direct way to determine whether investors anticipated a stock market crash using the implicit parameters above is to examine the jump expectations of the CVJD and SVJD models. While there is no universally accepted numerical definition of a stock market crash, it suffices for us to look for time periods between 2006 and 2010 in which heightened risk of significant negative, discontinuous index price movements are implicit in S&P 500 options. We will break the five-year period into three smaller segments for closer examination. Segment A (January 2006 to August 2008) covers a long period of steady increases in the index until its peak in October 2006, followed by a slow downturn including the major events in panel A of Table 1. Segment B (August 2008 to December 2008) includes the most drastic index downturns and rebounds that occurred during the financial crisis, including the events in panel B of Table 1. Segment C (April 2010 to June 2010) shows market reactions to the Flash Crash and other events in panel C of Table 1, followed by a return to the index’s slow and steady recovery. Figure 10A-10B shows the segmentations of the time series plots for the S&P 500 index and the implicit jump expectations from the SVJD and CVJD models. The vertical lines represent the midpoints of August 2008, December 2008, April 2010 and June 2010. Looking at the five years of data as a whole, we see no evidence that implied jump expectations anticipated the movements of the index—there do not appear to be any sustained periods of large negative jump expectations weeks or months before actual downturns in the index. However, in Figure 10B, the CVJD jump expectations (dashed line) take on large negative values in August 2007 and between February and March 2008. While these values seem disproportionate to the actual size of drops in the index, the spikes are not sustained. This may be better evidence for an overreaction in the change of options prices than for crash predictions. Interestingly, the spikes for the SVJD model (Figure
10B, solid line) sometimes seem to be lagging those in the plots for the CVJD model and the S&P 500 index in terms of negative jump expectations coinciding with declines in the index. This may be because the market pessimism implicit in options prices manifests as jump risk in the CVJD model, but manifests partially as heightened volatility and negative correlation between the volatility and the asset price in the SVJD model.

**Segment A, January 2006 to August 2008.** Figure 11A-11B below gives the times series plots for the index and implicit jump expectations from the SVJD and CVJD models for the period from January 3, 2006 to August 15, 2008, plotted on the same scales as Figure 10. Generally, the CVJD jump expectations plot (Figure 11B, dashed line) moves contemporaneously with the index and most of its moves are proportionate, indicating that there were no crash predictions or over-reactionary jump expectations during this period. The SVJD model’s jump expectations plot movements (Figure 11B, solid line) are not as well-synchronized with index price changes, and the sizes of the jumps are muted compared to both the movements in the index and the corresponding CVJD plot. We see no indications of crash predictions in the implicit SVJD parameters during this time period, but the jump parameters of both models do seem to respond to significant changes in the S&P price. On each of the highlighted dates, when the index drops by 2% or more, the CVJD model’s jump expectations also drop significantly, and the same is true for the SVJD model with the exception of March 14, 2008.

**Segment B, August 2008 to December 2008.** Figure 12A-12B below gives the times series plots for the index and implicit jump expectations from the SVJD and CVJD models for the period from August 15, 2008 to December 15, 2008. The CVJD model’s implicit jump expectations seem to predict a worst-case scenario (a crash), except that these negative jump expectations arise and recede as the index falls and rebounds—that is, the movements in Figure
12B appear to be responsive rather than predictive. The jump expectations go largely negative three days before Lehman Brothers declared bankruptcy and stayed largely negative for the rest of 2008, but the index itself was also declining for several days prior to the bankruptcy, and this shift in the CVJD plot may indicate more of an “awakening” of the market than a crash prediction.

The SVJD model’s implicit jump expectations (Figure 12B, solid line) remain largely unmoved during much of September and October before large negative swings at the end of October and end of November. These SVJD jump expectations do not seem to anticipate a crash, nor do they appear to react swiftly to downturns in the index, at least as compared to CVJD plot. Again, the magnitude of jump expectations implicit in the SVJD model are muted compared to the CVJD model, and this may be because downside risk is being modeled as enlarged volatility and continuous index declines rather than expected discontinuous price jumps.

**Segment C, April 2010 to June 2010.** The Flash Crash occurred on May 6, 2010 and while the individual stocks affected by the 2:45 computer glitch did recover, the index still dropped 3% on the day and fell another 2% the following day. The index almost completely recovered on May 10, and saw sharp declines later in May that were likely unrelated to the Flash Crash. Figure 13A-13B below gives the time series plots for the index and implicit jump expectations from the SVJD and CVJD models for the period from April 15, 2010 to June 15, 2010. The Flash Crash provides another opportunity to show crash expectations or persistent increased risk of negative price jumps that may be implicit in the options market, but both the CVJD and SVJD models’ jump expectations seem to move with the market rather than ahead of

---

it, and in relatively muted fashion compared to the parameter levels we see during the financial crisis. We see that jump expectations behave similarly in the days surrounding the unpredictable Flash Crash and during the financial crisis, which indicates that investors were reacting to the market movements of the financial crisis rather than anticipating them.
Conclusion

Of the four models applied to the data set of daily prices of options on the S&P 500 index, the SVJD model fit the best, in agreement with the findings by previous researchers this model typically has the best options pricing performance and the most internal consistency of these four models. Plots of implicit jump expectations from the CVJD and SVJD models indicated that investor sentiment as expressed by options prices was merely reactive to the events of the financial crisis (and Flash Crash), and was not an indicator of stock market crash predictions like Bates (1991) found before the 1987 crash. We see that market participants were optimistic and expecting continued prosperity throughout 2006, and that concern about the financial system began to rise in mid-2007, but that information implicit in options prices did not predict the kind of decline that marked the 2007-2009 financial crisis.
Appendix A: Figures

Figure 1. S&P 500 index prices, 2006-2010.

Figure 2. Black-Scholes implicit volatility: $\sigma$. 
Figure 3. CVJD parameters implied by European options on the S&P 500 index, 2006-2010. The plots in each panel are of: A) the implicit continuous volatility conditional on no price jumps occurring $\sigma$; B) the implicit standard deviation of jump sizes $\sigma_j$; C) the implicit jump volatility: $\sqrt{\lambda \left[ \ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2 \right]^2 + \sigma_j^2}$; D) the implicit spot volatility: $\sqrt{\sigma^2 + \lambda \left[ \ln(1 + \mu) - \frac{1}{2} \sigma^2 \right]^2 + \sigma^2}$ and the deviation of it from Black-Scholes implied volatility (dashed line); E) mean number of discontinuous price jumps per year $\bar{\lambda}$; and F) the mean jump size as a percentage of the asset price $\bar{\mu}_j$. 

36
Figure 3.—Continued
Figure 3.—Continued
Figure 4. CVJD jump expectations implied by European options on the S&P 500 index, 2006-2010:

\[ (1 + \mu_t)^{\lambda} - 1. \]
Figure 5. SV parameters implied by European options on the S&P 500 index, 2006-2010. The plots in each panel are of: A) the implicit spot volatility \( \sqrt{\sigma_s} \); B) the implicit long-run volatility \( \sqrt{\sigma_p} / \sqrt{\kappa} \); C) the implicit volatility of volatility \( \sigma_{\sigma_p} \); and D) the implicit correlation between the innovations in asset prices and volatility \( \rho \).
Figure 5.—Continued
Figure 6. Time series plot of 11-day centered moving average of SV implicit correlation parameter $\rho$. 
Figure 7. SVJD parameters implied by European options on the S&P 500 index, 2006-2010. The plots in each panel are of: A) the implicit continuous volatility $\sqrt{\nu_t}$; B) the implicit long-run continuous volatility $\sqrt{\nu_\infty}$; C) the implicit volatility of volatility $\sigma_\nu$; D) the implicit standard deviation of jump size $\sigma_j$; E) the implicit jump volatility $\sqrt{\frac{1}{2} \lambda \left[ \ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2 \right]^2 + \sigma_j^2}$; F) the implicit unconditional spot volatility $\sqrt{\nu + \lambda \left[ \ln(1 + \mu_j) - \frac{1}{2} \sigma_j^2 \right]^2 + \sigma_j^2}$ and the deviation from Black-Scholes (dashed line); G) the implicit correlation of the innovations in asset prices and continuous volatility $\rho$; H) the implicit annual mean number of discrete jumps $\lambda$; and I) the implicit mean jump size as a percentage of the asset price $\mu_j$. 
Figure 7.—Continued
Figure 7.—Continued
Figure 7.—Continued
Figure 7.—Continued
Figure 8. Time series plot of 11-day centered moving average of SVJD implicit correlation parameter $\rho$.

Figure 9. SVJD jump expectations implied by European options on the S&P 500 index, 2006-2010: $\left(1 + \mu_j\right)^2 - 1$. 
Figure 10. Segmenting the data for detailed analysis. The vertical dashed lines represent the midpoints of August 2008, December 2008, April 2008, and June 2008. The three segmentation graphs are for: A) the S&P 500 index; and B) implicit jump expectations from the SVJD model (solid line) and the CVJD model (dashed line).
Figure 11. Segment A: January 3, 2006 to August 15, 2008. The vertical dashed lines represent February 27th; June 7th; and July 31st, 2007; and March 14th, 2008, respectively. The time series plots are for: A) the S&P 500 index; and B) implicit jump expectations from the SVJD model (solid line) and the CVJD model (dashed line).
Figure 12. Segment B: August 15, 2008 to December 15, 2008. The vertical dashed lines represent September 15th, 17th, and 29th; October 9th, 13th, 15th, 28th; November 13th; and December 1st, 2008, respectively. The time series plots are for: A) the S&P 500 index; and B) implicit jump expectations from the SVJD model (solid line) and the CVJD model (dashed line).
Figure 13. Segment C: April 15, 2010 to June 10, 2010. The vertical dashed lines represent May 6th, 7th, 10th, 14th, and 20th, 2010, respectively. The time series plots are for: A) the S&P 500 index; and B) implicit jump expectations from the SVJD model (solid line) and the CVJD model (dashed line).
### Table 1. List of important dates for S&P 500 index.
Includes dates, index closing prices, and price changes in percentages and dollars for dates: (A) leading up to and beginning the financial crisis, (B) during the height of the financial crisis, and (C) surrounding the Flash Crash of May 2010.

<table>
<thead>
<tr>
<th>Date</th>
<th>S&amp;P 500 Index</th>
<th>Change (%)</th>
<th>Change ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/27/2007</td>
<td>$1,399.04</td>
<td>-3%</td>
<td>-$50.33</td>
</tr>
<tr>
<td>6/7/2007</td>
<td>$1,490.72</td>
<td>-2%</td>
<td>-$26.66</td>
</tr>
<tr>
<td>7/31/2007</td>
<td>$1,455.27</td>
<td>-1%</td>
<td>-$18.64</td>
</tr>
<tr>
<td>3/14/2008</td>
<td>$1,288.14</td>
<td>-2%</td>
<td>-$27.34</td>
</tr>
<tr>
<td>9/15/2008</td>
<td>$1,192.70</td>
<td>-5%</td>
<td>-$59.00</td>
</tr>
<tr>
<td>9/17/2008</td>
<td>$1,156.39</td>
<td>-5%</td>
<td>-$57.21</td>
</tr>
<tr>
<td>9/29/2008</td>
<td>$1,106.42</td>
<td>-9%</td>
<td>-$106.85</td>
</tr>
<tr>
<td>10/9/2008</td>
<td>$909.92</td>
<td>-8%</td>
<td>-$75.02</td>
</tr>
<tr>
<td>10/13/2008</td>
<td>$1,003.35</td>
<td>12%</td>
<td>$104.13</td>
</tr>
<tr>
<td>10/15/2008</td>
<td>$907.84</td>
<td>-9%</td>
<td>-$90.17</td>
</tr>
<tr>
<td>10/28/2008</td>
<td>$940.51</td>
<td>11%</td>
<td>$91.59</td>
</tr>
<tr>
<td>11/13/2008</td>
<td>$911.29</td>
<td>7%</td>
<td>$58.99</td>
</tr>
<tr>
<td>12/1/2008</td>
<td>$816.21</td>
<td>-9%</td>
<td>-$80.03</td>
</tr>
<tr>
<td>5/6/2010</td>
<td>$1,128.15</td>
<td>-3%</td>
<td>-$37.72</td>
</tr>
<tr>
<td>5/7/2010</td>
<td>$1,110.88</td>
<td>-2%</td>
<td>-$17.27</td>
</tr>
<tr>
<td>5/10/2010</td>
<td>$1,159.73</td>
<td>4%</td>
<td>$48.85</td>
</tr>
<tr>
<td>5/14/2010</td>
<td>$1,135.68</td>
<td>-2%</td>
<td>-$21.76</td>
</tr>
<tr>
<td>5/20/2010</td>
<td>$1,071.59</td>
<td>-4%</td>
<td>-$43.46</td>
</tr>
</tbody>
</table>
### Table 2. List of lower and upper bounds for parameter estimation algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Applicable Models</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>continuous volatility</td>
<td>B-S, CVJD, SVJD, SVJD</td>
<td>0.00001</td>
<td>1.5</td>
</tr>
<tr>
<td>$\nu_t$</td>
<td>instantaneous continuous variance</td>
<td>SV, SVJD</td>
<td>0.00001</td>
<td>1.5</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>mean number of jumps per year</td>
<td>CVJD, SVJD</td>
<td>0</td>
<td>160</td>
</tr>
<tr>
<td>$\mu_j$</td>
<td>mean jump size (percentage), given a jump occurs</td>
<td>CVJD, SVJD</td>
<td>-0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$\sigma_j$</td>
<td>jump volatility</td>
<td>CVJD, SVJD</td>
<td>0.00001</td>
<td>0.99</td>
</tr>
<tr>
<td>$\theta$</td>
<td>long-run variance parameters</td>
<td>SV, SVJD</td>
<td>0.00001</td>
<td>20</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>volatility of volatility</td>
<td>SV, SVJD</td>
<td>0.00001</td>
<td>2</td>
</tr>
<tr>
<td>$\rho$</td>
<td>correlation of asset price and continuous volatility</td>
<td>SV, SVJD</td>
<td>-0.99999</td>
<td>0.99999</td>
</tr>
</tbody>
</table>

### Table 3. Pricing errors for all four models. Based on calculated prices for each of the 262,962 options using the appropriate trading day’s estimated parameters.

<table>
<thead>
<tr>
<th>Error Measure</th>
<th>B-S</th>
<th>CVJD</th>
<th>SV</th>
<th>SVJD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Absolute Price Error</td>
<td>$\text{3.20}$</td>
<td>$\text{1.67}$</td>
<td>$\text{2.24}$</td>
<td>$\text{1.55}$</td>
</tr>
<tr>
<td>Maximum Absolute Price Error</td>
<td>$\text{9.40}$</td>
<td>$\text{3.91}$</td>
<td>$\text{6.43}$</td>
<td>$\text{3.53}$</td>
</tr>
<tr>
<td>Average Absolute Percentage Error</td>
<td>38.27 %</td>
<td>29.39 %</td>
<td>29.70 %</td>
<td>13.45 %</td>
</tr>
<tr>
<td>Maximum Absolute Percentage Error</td>
<td>144.99 %</td>
<td>138.78 %</td>
<td>181.62 %</td>
<td>66.64 %</td>
</tr>
<tr>
<td>Sum of Squared Price Errors</td>
<td>3,492,907</td>
<td>962,825</td>
<td>1,672,028</td>
<td>684,975</td>
</tr>
</tbody>
</table>
Appendix C: Matlab Code

The following is the code for Matlab’s “lsqcurvefit.m” program, which was used for least-squares estimation of daily parameters, as described on page 23 above. Comment lines begin with “%”.

```matlab
function [xCurrent,Resnorm,FVAL,EXITFLAG,OUTPUT,LAMBDA,JACOB] = lsqcurvefit(FUN,xCurrent,XDATA,YDATA,LB,UB,options,varargin)

%LSQCURVEFIT solves non-linear least squares problems.
% LSQCURVEFIT attempts to solve problems of the form:
%   min  sum {(FUN(X,XDATA)-YDATA).^2}  where X, XDATA, YDATA and the
%   X values returned by FUN can be vectors or matrices.
%
% LSQCURVEFIT implements two different algorithms: trust region reflective and
% Levenberg-Marquardt. Choose one via the option Algorithm; for instance, to
% choose Levenberg-Marquardt, set OPTIONS = optimset('Algorithm','levenberg-marquardt'),
% and then pass OPTIONS to LSQCURVEFIT.
%
% X = LSQCURVEFIT(FUN,X0,XDATA,YDATA) starts at X0 and finds coefficients
% X to best fit the nonlinear functions in FUN to the data YDATA (in the
% least-squares sense). FUN accepts inputs X and XDATA and returns a
% vector (or matrix) of function values F, where F is the same size as
% YDATA, evaluated at X and XDATA. NOTE: FUN should return FUN(X,XDATA)
% and not the sum-of-squares sum((FUN(X,XDATA)-YDATA).^2).
% ((FUN(X,XDATA)-YDATA) is squared and summed implicitly in the
% algorithm.)
%
% X = LSQCURVEFIT(FUN,X0,XDATA,YDATA,LB,UB) defines a set of lower and
% upper bounds on the design variables, X, so that the solution is in the
% range LB <= X <= UB. Use empty matrices for LB and UB if no bounds
% exist. Set LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if
% X(i) is unbounded above.
%
% X = LSQCURVEFIT(FUN,X0,XDATA,YDATA,LB,UB,OPTIONS) minimizes with the
% default parameters replaced by values in the structure OPTIONS, an
% argument created with the OPTIMSET function. See OPTIMSET for details.
% Use the Jacobian option to specify that FUN also returns a second output
% argument J that is the Jacobian matrix at the point X. If FUN returns
% a vector F of m components when X has length n, then J is an m-by-n matrix
% where J(i,j) is the partial derivative of F(i) with respect to x(j).
% (Note that the Jacobian J is the transpose of the gradient of F.)
%
% X = LSQCURVEFIT(PROBLEM) solves the non-linear least squares problem
% defined in PROBLEM. PROBLEM is a structure with the function FUN in
```

55
% PROBLEM.objective, the start point in PROBLEM.x0, the 'xdata' in
% PROBLEM.xdata, the 'ydata' in PROBLEM.ydata, the lower bounds in
% PROBLEM.lb, the upper bounds in PROBLEM.ub, the options structure in
% PROBLEM.options, and solver name 'lsqcurvefit' in PROBLEM.solver. Use
% this syntax to solve at the command line a problem exported from
% OPTIMTOOL. The structure PROBLEM must have all the fields.
%
% [X,RESNORM] = LSQCURVEFIT(FUN,X0,XDATA,YDATA,...) returns the value of
% the squared 2-norm of the residual at X: sum \{(FUN(X,XDATA)-YDATA).^2\}.
%
% [X,RESNORM,RESIDUAL] = LSQCURVEFIT(FUN,X0,...) returns the value of
% residual, FUN(X,XDATA)-YDATA, at the solution X.
%
% [X,RESNORM,RESIDUAL,EXITFLAG] = LSQCURVEFIT(FUN,X0,XDATA,YDATA,...)
% returns an EXITFLAG that describes the exit condition of LSQCURVEFIT.
% Possible values of EXITFLAG and the corresponding exit conditions are
% listed below. See the documentation for a complete description.
%
% 1  LSQCURVEFIT converged to a solution.
% 2  Change in X too small.
% 3  Change in RESNORM too small.
% 4  Computed search direction too small.
% 0  Too many function evaluations or iterations.
% -1  Stopped by output/plot function.
% -2  Bounds are inconsistent.
%
% [X,RESNORM,RESIDUAL,EXITFLAG,OUTPUT] = LSQCURVEFIT(FUN,X0,XDATA,
% YDATA,...) returns a structure OUTPUT with the number of iterations
% taken in OUTPUT.iterations, the number of function evaluations in
% OUTPUT.funcCount, the algorithm used in OUTPUT.algorithm, the number of
% CG iterations (if used) in OUTPUT.cgiterations, the first-order
% optimality (if used) in OUTPUT.firstorderopt, and the exit message in
% OUTPUT.message.
%
% [X,RESNORM,RESIDUAL,EXITFLAG,OUTPUT,LAMBDA] =
% LSQCURVEFIT(FUN,X0,XDATA,
% YDATA,...) returns the set of Lagrangian multipliers, LAMBDA, at the
% solution: LAMBDA.lower for LB and LAMBDA.upper for UB.
%
% [X,RESNORM,RESIDUAL,EXITFLAG,OUTPUT,LAMBDA,JACOBIAN] =
% LSQCURVEFIT(FUN,
% X0,XDATA,YDATA,...) returns the Jacobian of FUN at X.
%
% Examples
% FUN can be specified using @:
% xdata = [5;4;6]; % example xdata
ydata = 3*sin([5;4;6])+6; % example ydata
x = lsqcurvefit(@myfun, [2 7], xdata, ydata)

where myfun is a MATLAB function such as:

function F = myfun(x,xdata)
F = x(1)*sin(xdata)+x(2);

FUN can also be an anonymous function:
x = lsqcurvefit(@(x,xdata) x(1)*sin(xdata)+x(2),[2 7],xdata,ydata)

If FUN is parameterized, you can use anonymous functions to capture the problem-dependent parameters. Suppose you want to solve the curve-fitting problem given in the function myfun, which is parameterized by its second argument c. Here myfun is a MATLAB file function such as

function F = myfun(x,xdata,c)
F = x(1)*exp(c*xdata)+x(2);

To solve the curve-fitting problem for a specific value of c, first assign the value to c. Then create a two-argument anonymous function that captures that value of c and calls myfun with three arguments. Finally, pass this anonymous function to LSQCURVEFIT:

xdata = [3; 1; 4]; % example xdata
ydata = 6*exp(-1.5*xdata)+3; % example ydata
c = -1.5; % define parameter
x = lsqcurvefit(@(x,xdata) myfun(x,xdata,c),[5;1],xdata,ydata)

See also OPTIMSET, LSQNONLIN, FSOLVE, @, INLINE.

Copyright 1990-2011 The MathWorks, Inc.
$Revision: 1.1.6.10 $  $Date: 2011/07/31 13:15:53 $
'Jacobian', 'off', ...
'JacobMult', [], ...
'JacobPattern', 'sparse(ones(Jrows, Jcols))', ...
'MaxFunEvals', [], ...
'MaxIter', 400, ...
'MaxPCGIter', 'max(1, floor(numberOfVariables/2))', ...
'OutputFcn', [], ...
'PlotFcns', [], ...
'PrecondBandWidth', Inf, ...
'ScaleProblem', 'none', ...
'TolFun', 1e-6, ...
'TolPCG', 0.1, ...
'TolX', 1e-6, ...
'TypicalX', 'ones(numberOfVariables, 1)');

% If just 'defaults' passed in, return the default options in X
if nargin == 1 && nargout <= 1 && isequal(FUN, 'defaults')
  xCurrent = defaultopt;
  return
end

if nargin < 7
  options = [];
  if nargin < 6
    UB = [];
    if nargin < 5
      LB = [];
    end
  end
end

problemInput = false;
if nargin == 1
  if isa(FUN, 'struct')
    problemInput = true;
    [FUN, xCurrent, XDATA, YDATA, LB, UB, options] = separateOptimStruct(FUN);
  else
    error(message('optimlib:lsqcurvefit:InputArg'));
  end
end

if nargin < 4 && ~problemInput
  error(message('optimlib:lsqcurvefit:NotEnoughInputs'));
end

% Check for non-double inputs
SUPERIORFLOAT errors when superior input is neither single nor double;
We use try-catch to override SUPERIORFLOAT's error message when input
data type is integer.

```
try
    dataType = superiorfloat(xCurrent,YDATA,LB,UB);
catch ME
    if strcmp(ME.identifier,'MATLAB:datatypes:superiorfloat')
        dataType = 'notDouble';
    end
end
```

if ~strcmp(dataType,'double')
    error(message('optimlib:lsqcurvefit:NonDoubleInput'))
end

caller = 'lsqcurvefit';
[funfcn_x_xdata,mtxmpy_xdata,flags,sizes,funValCheck,xstart,lb,ub,EXITFLAG, ...
    Resnorm,FVAL,LAMBDA,JACOB,OUTPUT,earlyTermination] = ...
    lsqnsetup(FUN,xCurrent,LB,UB,options,defaultopt,caller,nargout,length(varargin));
if earlyTermination
    return % premature return because of problem detected in lsqnsetup()
end

```
xCurrent() = xstart; % reshape back to user shape before evaluation
funfcn = funfcn_x_xdata; % initialize user functions funfcn, which depend only on x
% Catch any error in user objective during initial evaluation only
switch funfcn_x_xdata{1}
    case 'fun'
        try
            initVals.F = feval(funfcn_x_xdata{3},xCurrent,XDATA,varargin{:});
catch userFcn_ME
            optim_ME = MException('optimlib:lsqcurvefit:InvalidFUN', ...
                getString(message('optimlib:lsqcurvefit:InvalidFUN')));
            userFcn_ME = addCause(userFcn_ME,optim_ME);
            rethrow(userFcn_ME)
        end
        initVals.J = [];
        funfcn{3} = @objective;
    case 'fungrad'
        try
            [initVals.F,initVals.J] = feval(funfcn_x_xdata{3},xCurrent,XDATA,varargin{:});
catch userFcn_ME
            optim_ME = MException('optimlib:lsqcurvefit:InvalidFUN', ...
                getString(message('optimlib:lsqcurvefit:InvalidFUN')));
            userFcn_ME = addCause(userFcn_ME,optim_ME);
            rethrow(userFcn_ME)
        end
```

59
funfcn{3} = @objectiveAndJacobian;
funfcn{4} = @objectiveAndJacobian;
case 'fun_then_grad'
    try
        initVals.F = feval(funfcn_x_xdata{3},xCurrent,XDATA,varargin{:});
        catch userFcn_ME
            optim_ME = MException('optimlib:lsqcurvefit:InvalidFUN', ...
                    getString(message('optimlib:lsqcurvefit:InvalidFUN')));
            userFcn_ME = addCause(userFcn_ME,optim_ME);
            rethrow(userFcn_ME)
        end
    try
        initVals.J = feval(funfcn_x_xdata{4},xCurrent,XDATA,varargin{:});
        catch userFcn_ME
            optim_ME = MException('optimlib:lsqcurvefit:InvalidFUNJac', ...
                    getString(message('optimlib:lsqcurvefit:InvalidFUNJac')));
            userFcn_ME = addCause(userFcn_ME,optim_ME);
            rethrow(userFcn_ME)
        end
    funfcn{3} = @objective;
    funfcn{4} = @jacobian;
    otherwise
        error(message('optimlib:lsqcurvefit:UndefCallType'))
    end

if ~isequal(size(initVals.F),size(YDATA))
    error(message('optimlib:lsqcurvefit:YdataSizeMismatchFunVal'))
end
initVals.F = initVals.F - YDATA; % preserve initVals.F shape until after subtracting YDATA

% Pass functions that depend only on x: funfcn and jacobmult
[xCurrent,Resnorm,FVAL,EXITFLAG,OUTPUT,LAMBDA,JACOB] = ...
    lsqncommon(funfcn,xCurrent,lb,ub,options,defaultopt,caller,...
            initVals,sizes,flags,@jacobmult,varargin{:});

% Nested functions that depend only on x and capture the constant values
% xdata and ydata, and also varargin
function F = objective(x,varargin)
    F = feval(funfcn_x_xdata{3},x,XDATA,varargin{:});
    F = F - YDATA;
end
function [F,J] = objectiveAndJacobian(x,varargin)
    % Function value and Jacobian returned together
    [F,J] = feval(funfcn_x_xdata{3},x,XDATA,varargin{:});
    F = F - YDATA;
end
function J = jacobian(x, varargin)
    % Jacobian returned in separate function
    J = feval(funfcn_x_xdata{4}, x, XDATA, varargin{:});
end
function W = jacobmult(Jinfo, Y, flag, varargin)
    W = feval(mtxmpy_xdata, Jinfo, Y, flag, XDATA, varargin{:});
end
end
The following is the code for Matlab’s “integral.m” program, which was used for numerical evaluation of integrals via Gaussian quadrature, as described on page 23 above. Comment lines begin with “%”.

function Q = integral(fun,a,b,varargin)
%INTEGRAL  Numerically evaluate integral.
% Q = INTEGRAL(FUN,A,B) approximates the integral of function FUN from A
% to B using global adaptive quadrature and default error tolerances.
%
% FUN must be a function handle. A and B can be -Inf or Inf. If both are
% finite, they can be complex. If at least one is complex, INTEGRAL
% approximates the path integral from A to B over a straight line path.
%
% For scalar-valued problems the function Y = FUN(X) must accept a vector
% argument X and return a vector result Y, the integrand function
% evaluated at each element of X. For array-valued problems (see the
% 'ArrayValued' option below) FUN must accept a scalar and return an
% array of values.
%
% Q = INTEGRAL(FUN,A,B,PARAM1,VAL1,PARAM2,VAL2,...) performs the
% integration with specified values of optional parameters. The available
% parameters are
%
% 'AbsTol', absolute error tolerance
% 'RelTol', relative error tolerance
%
% INTEGRAL attempts to satisfy |Q - I| <= max(AbsTol,RelTol*|Q|),
% where I denotes the exact value of the integral. Usually RelTol
% determines the accuracy of the integration. However, if |Q| is
% sufficiently small, AbsTol determines the accuracy of the
% integration, instead. The default value of AbsTol is 1.e-10, and
% the default value of RelTol is 1.e-6. Single precision integrations
% may require larger tolerances.
%
% 'ArrayValued', FUN is an array-valued function when the input is scalar
%
% When 'ArrayValued' is true, FUN is only called with scalar X, and
% if FUN returns an array, INTEGRAL computes a corresponding array of
% outputs Q. The default value is false.
%
% 'Waypoints', vector of integration waypoints
%
% If FUN(X) has discontinuities in the interval of integration, the
% locations should be supplied as a 'Waypoints' vector. Waypoints
% should not be used for singularities in FUN(X). Instead, split the
% interval and add the results from separate integrations with
singularities at the endpoints. If A, B, or any entry of the
waypoints vector is complex, the integration is performed over a
sequence of straight line paths in the complex plane, from A to the
first waypoint, from the first waypoint to the second, and so
forth, and finally from the last waypoint to B.

Examples:

% Integrate f(x) = exp(-x^2)*log(x)^2 from 0 to infinity:
f = @(x) exp(-x.^2).*log(x).^2
Q = integral(f,0,Inf)

% To use a parameter in the integrand:
f = @(x,c) 1./(x.^3-2*x-c)
Q = integral(@(x)f(x,5),0,2)

% Specify tolerances:
Q = integral(@(x)log(x),0,1,'AbsTol',1e-6,'RelTol',1e-3)

% Integrate f(z) = 1/(2z-1) in the complex plane over the
% triangular path from 0 to 1+1i to 1-1i to 0:
Q = integral(@(z)1./(2*z-1),0,0,'Waypoints',[1+i,1-i])

% Integrate the vector-valued function sin((1:5)*x) from 0 to 1:
Q = integral(@(x)sin((1:5)*x),0,1,'ArrayValued',true)

Class support for inputs A, B, and the output of FUN:
float: double, single

See also INTEGRAL2, INTEGRAL3, FUNCTION_HANDLE

Portions based on "quadva" by Lawrence F. Shampine.
Ref: L.F. Shampine, "Vectorized Adaptive Quadrature in Matlab",
Journal of Computational and Applied Mathematics 211, 2008, pp.131-140

Copyright 2007-2011 The MathWorks, Inc.
$Revision: 1.1.6.2 $ $Date: 2011/11/13 04:01:00 $

Validate the first three inputs.
narginchk(3,inf);
if ~isa(fun,'function_handle')
    error(message('MATLAB:integral:funArgNotHandle'));
end
if ~(isscalar(a) && isfloat(a) && isscalar(b) && isfloat(b))
    error(message('MATLAB:integral:invalidEndpoint'));
end
opstruct = integralParseArgs(varargin{:});
Q = integralCalc(fun,a,b,opstruct);
References


