Fast Monte Carlo Valuation of American Options with Python & Numpy

Dr. Yves J. Hilpisch

27 August 2011

Paris, EuroScipy 2011

Visixion GmbH
Derivatives Analytics & Python Programming
1 Market Realities
   - Index Returns
   - Implied Volatility Surface
   - Interest Rates

2 The SVSI Model
   - Model Economy
   - European Options
   - Monte Carlo Simulation

3 Least Squares Monte Carlo
   - American Options
   - Variance Reduction

4 Numerical Results
   - Example Options & Accuracy
   - Selected Results

5 Conclusions

6 Contact
Guiding Questions for the Talk

1. Why do we need a stochastic volatility, stochastic short rate (SVSI) model at all?
2. How do I value American options via Monte Carlo simulation?
   ▶ problem: Monte Carlo is forward evolving, American options are valued backwards
3. How do I perform a Monte Carlo simulation for a SVSI model?
4. How can I improve performance of the Monte Carlo valuation of American options in such a model?
5. What are reasonable practical requirements with regard to valuation accuracy?
6. How fast can we value American put options accurately in the SVSI model (with Python)?
   ▶ benchmark: “dozens of minutes” in Matlab environment
BASIC NOTIONS: historical/realized, instantaneous, implied volatility

- **historical/realized volatility**: the standard deviation of log returns of a financial time series; assume \( N \) (past) log returns \( r_n \equiv \log S_n - \log S_{n-1}, n \in \{1, ..., N\} \), with mean return \( \hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} r_n \); the historical/realized volatility \( \hat{\sigma} \) is then given by

\[
\hat{\sigma} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (r_n - \hat{\mu})^2}
\]

- **instantaneous volatility**: the volatility factor of a diffusion process; for example, in the Black-Scholes-Merton model the instantaneous volatility \( \sigma \) is found in the respective SDE

\[
dS_t = rS_t dt + \sigma S_t dZ_t
\]

- **implied volatility**: the volatility that, if put into the Black-Scholes-Merton option pricing formula, gives the market-observed price of an option; suppose we observe a price of \( C^*_0 \) for a European call option; the implied volatility \( \sigma^{imp} \) is the quantity that solves ceteris paribus the implicit equation

\[
C^*_0 = C^{BSM}(S_0, K, T, r, \sigma^{imp})
\]
DAX index returns—volatility clustering¹

¹Source: http://finance.yahoo.com, 29 Apr 2011
DAX volatility—stochastic & mean reverting volatility, leverage effect²

²Source: http://finance.yahoo.com, 29 Apr 2011
DAX implied volatility surface—smiles and term structure

maturities: 21 (red dots), 49 (green crosses), 140 (blue triangles), 231 (yellow stones) and 322 days (purple hectagons)

Source: http://www.eurexchange.com, 29 Apr 2011
EURIBOR 1 week—positivity, stochasticity, mean reversion, clustering

Source: http://www.euribor-ebf.eu, 02 May 2011
EURIBOR—term structure

EURIBOR 1 week (red line), 1 month (blue dashed line), 6 months (green dash-dotted line) and 1 year (magenta dotted line)

Source: [http://www.euribor-ebf.eu](http://www.euribor-ebf.eu), 02 May 2011
A realistic market model ...

- ... has to take into account that index volatility
  - ... varies over time (stochasticity, mean reversion, clustering)
  - ... is negatively correlated with returns (leverage effect)
  - ... varies for different option strikes (volatility smile)
  - ... varies for different option maturities (volatility term structure)

- ... has to take into account that interest rates
  - ... vary over time (positivity, stochasticity, mean reversion)
  - ... vary for different time horizons (term structure)

- ... therefore comprises
  - ... a stochastic volatility component and
  - ... a stochastic short rate component
Model Economy—stochastic volatility model

- given is a filtered probability space $\{\Omega, \mathcal{F}, \mathbb{F}, P\}$ representing uncertainty in the model economy with final date $0 < T < \infty$
- MS2009 consider the stochastic volatility model of Heston (1993, H93) with stochastic short rate model of Cox, Ingersoll and Ross (1985, CIR85)
- the stochastic volatility part in risk-neutral form is given by the SDEs

$$dS_t = r_t S_t dt + \sqrt{v_t} S_t dZ^1_t$$  (1)

$$dv_t = \kappa_v (\theta_v - v_t) dt + \sigma_v \sqrt{v_t} dZ^2_t$$  (2)

for $0 \leq t \leq T$ and with the following meaning of the variables and parameters

- $S_t$ index level at date $t$
- $r_t$ risk-less short rate at date $t$
- $v_t$ variance at date $t$
- $\kappa_v$ speed of adjustment of $v_t$ to ...
- $\theta_v$, the long-term average of the variance
- $\sigma_v$ volatility coefficient of the index’s variance
- $Z^1_t$ standard Brownian motions
The SVSI Model

Model Economy—stochastic short rate model

- the stochastic short rate part of the model according to CIR85 is given by the SDE

\[ dr_t = \kappa_r (\theta_r - r_t) dt + \sigma_r \sqrt{r_t} dZ^3_t \]  

(3)

where the variables and parameters are defined as

- \( r_t \) short rate at date \( t \)
- \( \kappa_r \) speed of adjustment of \( r_t \) to ...
- \( \theta_r \), the long-term average of the short rate
- \( \sigma_r \) volatility coefficient of the short rate
- \( Z^3_t \) standard Brownian motion

- all processes are adapted to the filtration \( \mathbb{F} \)
- correlations are given by \( dZ^1_t dZ^2_t \equiv \rho dt \), \( dZ^1_t dZ^3_t \equiv dZ^2_t dZ^3_t \equiv 0 \), i.e. the short rate process is neither correlated with the index level nor the variance
- the time \( t \) value of a zero-coupon bond paying one unit of currency at \( T, 0 \leq t < T \), is

\[ B_t(T) = E_t^Q \left( \exp \left( - \int_t^T r_u du \right) \right) \]

- we define the set of uncertainties by \( X_t \equiv (S_t, v_t, r_t) \)
European Options—risk-neutral valuation

- by the **Fundamental Theorem of Asset Pricing**, the time \( t \) value of an attainable/redundant and \( \mathcal{F}_T \)–measurable contingent claim \( V_T \equiv h_T(X_T) \geq 0 \) (satisfying suitable integrability conditions) is given by arbitrage as

\[
V_t = \mathbb{E}_t^Q (B_t(T) V_T)
\]

with \( V_0 = \mathbb{E}_0^Q (B_0(T) V_T) \) as the important special case for valuation purposes

- \( \mathbb{E} \) denotes the expectation operator\(^6\) and \( Q \) the unique risk-neutral probability measure equivalent to the real world measure \( P \)

- for example, the Black-Scholes-Merton model is known to be complete from which uniqueness of the risk-neutral measure \( Q \) follows

- the defining characteristic of \( Q \) is that it makes the discounted index level process a martingale

---

\(^6\) \( \mathbb{E}_t(\cdot) \) is short for the conditional expectation \( \mathbb{E}(\cdot|\mathcal{F}_t) \)
European Options—Fourier-based pricing

- The values of European calls and puts in the model of MS2009 are known in semi-analytic form by Fourier transform techniques and characteristic functions.
- Since the short rate part of the model is not correlated with the stochastic volatility index part, it suffices to consider the characteristic function $\phi_{093}^H$ of the Heston (1993) model (see equation (10)).
- Equipped with the characteristic function, the following Fourier-based formula of Lewis (2001) to price European call options with strike $K$ and maturity date $T$ can be applied:

$$C_{093}^{H93}(K, T) = S_0 - B_0(T) \frac{\sqrt{S_0K}}{\pi} \int_0^\infty \text{Re} \left[ e^{-iku} \phi_{093}^H(u - i/2, T) \right] \frac{du}{u^2 + 1/4}$$

where $k \equiv \log(S_0/K)$.
- The price of the corresponding put follows from put-call-parity as

$$P_{093}^{H93}(K, T) = C_{093}^{H93}(K, T) + B_0(T)K - S_0$$

- The discount factor $B_0(T)$ is known in closed form for the CIR85 model (see equation (11)).
Monte Carlo Simulation—Euler scheme

- to simulate the financial model, i.e. to generate numerical values for $X_t$, it has to be discretized
- to this end, divide the given time interval $[0, T]$ in equidistant sub-intervals $\Delta t$ such that now $t \in \{0, \Delta t, 2\Delta t, \ldots, T\}$, i.e. there are $M + 1$ points in time with $M \equiv T / \Delta t$
- a discretization of the continuous time market model (1), (2) and (3) of MS2009 is then given by the so called “Full Truncation” scheme (with $s = t - \Delta t$):

$$
S_t = S_s \exp((\bar{r}_t - v_t/2)\Delta t + \sqrt{v_t}\sqrt{\Delta t}z_t^1)
$$

$$
\tilde{v}_t = \tilde{v}_s + \kappa_v(\theta_v - \tilde{v}_s^+)\Delta t + \sigma_v \sqrt{\tilde{v}_s^+} \sqrt{\Delta t}z_t^2
$$

$$
v_t = \tilde{v}_t^+
$$

$$
\tilde{r}_t = \tilde{r}_s + \kappa_r(\theta_r - \tilde{r}_s^+)\Delta t + \sigma_r \sqrt{\tilde{r}_s^+} \sqrt{\Delta t}z_t^3
$$

$$
r_t = \tilde{r}_t^+
$$

with $\bar{r}_t \equiv (r_t^+ + r_s^+)/2$ and the $z_t^n$ being standard normal
- here, the $z_t^1$ and $z_t^2$ are correlated with $\rho$ while all other random variables are uncorrelated
American Options—risk-neutral valuation (I)

- the valuation of options with American, i.e. early, exercise can be formulated as an optimal stopping problem

\[
V_0 = \sup_{\tau \in [0,T]} \mathbb{E}_0^Q (B_0(\tau) h_\tau(S_\tau))
\]  

- definitions are:
  - $V_0$ present value of the American derivative
  - $\tau$ a $\mathbb{F}$—adapted stopping time
  - $T$ date of maturity
  - $B_0(\tau)$ discount factor appropriate for stopping time $\tau$
  - $h_\tau$ a non-negative, $\mathcal{F}_\tau$—measurable payoff function
  - $S_\tau$ the index level process stopped at $t = \tau$

- the expectation is again taken under the unique risk-neutral measure $Q$ (e.g. given by some kind of model calibration procedure)
American Options—risk-neutral valuation (II)

- To value American options by Monte Carlo simulation (MCS), the optimal stopping problem (6) also has to be discretized:

\[ V_0 = \sup_{\tau \in \{0, \Delta t, 2\Delta t, \ldots, T\}} \mathbb{E}_0^Q (B_0(\tau) h_\tau(X_\tau)) \] (7)

- The continuation value \( C_t \) at date \( t \) of the option, i.e. the value of not exercising the option at this date, is given under risk-neutrality as

\[ C_t(x) = \mathbb{E}_t^Q \left( e^{\bar{r}_t \Delta t} V_{t+\Delta t}(X_{t+\Delta t}) | X_t = x \right) \]

using the Markov property of \( X_t \)

- Applying another important result in this context, the value of the American option at date \( t \) is then

\[ V_t(x) = \max[h_t(x), C_t(x)] \] (8)

i.e. the maximum of the payoff \( h_t(x) \) of immediate exercise and the expected payoff \( C_t(x) \) of not exercising.
The Algorithm—the steps and iterations

- simulate $I$ paths of $X$ with $M + 1$ points in time leading to values $X_{t,i}, t \in \{0, \ldots, T\}, i \in \{1, \ldots, I\}$

- **LSM algorithm of Longstaff and Schwartz (2001):** estimate the continuation values $C_{t,i}$ by ordinary least-squares regression—given the $I$ simulated values $X_{t,i}$ and continuation values $Y_{t,i}$ (use cross section of simulated data at date $t$)

- for $t = T$ the option value is $V_{T,i} = h_T(X_{T,i})$ by arbitrage

- start iterating backwards $t = T - \Delta t, \ldots, 0$:
  - regress the $Y_{t,i}$ against the $X_{t,i}, i \in \{1, \ldots, I\}$, given $D$ basis functions $b$
  - approximate $C_{t,i}$ by the resulting regression estimate $\hat{C}_{t,i}$ from previous regression step
  - according to (8) set
    \[
    V_{t,i} = \begin{cases} 
    h_t(X_{t,i}) & \text{if } h_t(X_{t,i}) > \hat{C}_{t,i} \text{ (exercise takes place)} \\
    Y_{t,i} & \text{if } h_t(X_{t,i}) \leq \hat{C}_{t,i} \text{ (no exercise takes place)}
    \end{cases}
    \]

- repeat iteration steps until $t = 0$

- for $t = 0$ calculate the LSM estimator

\[
\hat{V}_0^{LSM} = \frac{1}{I} \sum_{i=1}^{I} V_{0,i}
\]
The Algorithm—regression in action for put option
The Algorithm—backwards exercise/valuation in action for put option
Monte Carlo Simulation—variance reduction (VR) by control variates

- we correct the LSM estimator (9) by the simulated differences gained from a control variate
- consider that we have simulated $I$ paths of $X_t$ to value an American put option with maturity $T$ and strike $K$
- then there are also $I$ simulated present values of the corresponding European put option
- they are given by $P_{0,i} = B_0(T)h_T(S_{T,i}), i \in \{1, ..., I\}$, with $h_T(s) \equiv \max[K - s, 0]$
- we correct the estimator (9) as follows

$$\hat{V}_0^{CV} = \frac{1}{I} \sum_{i=1}^{I} \left( V_{0,i} - \lambda \cdot (P_{0,i} - P_0^{H93}) \right)$$

where we set for simplicity $\lambda \equiv 1$ and where we get $P_0^{H93}$ from (4) and (5)
Monte Carlo Simulation—VR by antithetic variates and moment matching

Matching of 1. and 2. moment of random numbers:

```python
# Function for Random Numbers
def RNG(M, I):
    if antiPaths == True:
        randDummy = standard_normal((3, M+1, I/2))
        rand = concatenate((randDummy, -randDummy), 2)
    else:
        rand = standard_normal((3, M+1, I))
    if moMatch == True:
        rand = rand / std(rand)
        rand = rand - mean(rand)
    return rand
```

Matching of 1. moment for index level dynamics:

```python
# Function for Heston Index Process
def eulerSExp(r, S0, v, row, CM):
    S = zeros((M+1, I), 'd')
    S[0, :] = S0
    bias = 0.0
    for t in range(1, M+1, 1):
        ran = dot(CM, rand[:, t, :])
        if moMatch == True:
            bias = mean(sqrt(v[t, :]) * ran[row] * sqrt(dt))
        S[t, :] = S[t-1, :] * exp(((r[t, :] + r[t-1, :]) / 2 - v[t, :] / 2) * dt) +
                  sqrt(v[t, :]) * ran[row] * sqrt(dt) - bias)
    return S
```
The Challenge

Medvedev and Scaillet (2009) write on page 16:

“To give an idea of the computational advantage of our method, a Matlab code implementing the algorithm of Longstaff and Schwartz (2001) takes dozens of minutes to compute a single option price while our approximation takes roughly a tenth of a second.”
Numerical Results—put options and accuracy considerations

- for a total of four parameter sets, American put options for three different maturities and moneyness levels, respectively, are valued:
  - $T \in \{ \frac{1}{12}, \frac{1}{4}, \frac{1}{2} \}$
  - $K \in \{90, 100, 110\}$
  - this makes for 36 American put options in total
- we say that our value estimate is reasonably accurate if either the absolute difference is
  - smaller than $PY_1$ currency units or
  - smaller than $PY_2$ percent.
- the first yardstick should apply to options with very small values (where the absolute difference is better suited) while the second rather applies to those with higher values (where the relative difference is better suited)
- In line with market microstructure benchmarks we set
  - $PY_1 = 0.025$ currency units (i.e. 2.5 cents) and
  - $PY_2 = 0.015$ (i.e. 1.5%).
EXCURSION: bid/ask spreads in options markets

Table: Option quote spreads for put options on stocks of the DJIA index

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Number</th>
<th>Maturity</th>
<th>Mid-Price</th>
<th>Spread</th>
<th>Rel. Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>All</td>
<td>1,105,028</td>
<td>96.07</td>
<td>5.093</td>
<td>0.229</td>
<td>7.80%</td>
</tr>
<tr>
<td>Short</td>
<td>OTM</td>
<td>158,486</td>
<td>44.55</td>
<td>1.339</td>
<td>0.148</td>
<td>15.98%</td>
</tr>
<tr>
<td>Short</td>
<td>ATM</td>
<td>120,257</td>
<td>44.63</td>
<td>3.443</td>
<td>0.204</td>
<td>7.12%</td>
</tr>
<tr>
<td>Short</td>
<td>ITM</td>
<td>146,979</td>
<td>43.86</td>
<td>6.858</td>
<td>0.279</td>
<td>4.91%</td>
</tr>
<tr>
<td>Long</td>
<td>OTM</td>
<td>267,847</td>
<td>128.80</td>
<td>2.238</td>
<td>0.172</td>
<td>10.26%</td>
</tr>
<tr>
<td>Long</td>
<td>ATM</td>
<td>201,100</td>
<td>129.33</td>
<td>5.769</td>
<td>0.255</td>
<td>5.18%</td>
</tr>
<tr>
<td>Long</td>
<td>ITM</td>
<td>210,359</td>
<td>127.34</td>
<td>10.621</td>
<td>0.317</td>
<td>3.50%</td>
</tr>
</tbody>
</table>

Data for the period 1996–2010; OTM, ATM, ITM = out-of-the, at-the, in-the-money options; number = number of put options included in the sample; maturity = average option maturity in days; mid-price = middle of bid and ask quotes in USD; spread = USD difference of bid and ask quote; relative spread = spread relative to mid-price. Source: Chaudhury (2011).

The tick size—i.e. the minimum allowed change of the price of an option—in the table for options with bid quotes below 3 USD is **5 cents**. For options with bid quotes above 3 USD the minimum tick size is **10 cents**.
Numerical Results—put option values and parameterizations

```python
# 'True' American Options Prices by Monte Carlo
# from MS (2009), table 3
trueList = array([(0.0001, 1.0438, 9.9950, 0.0346, 1.7379, 9.9823,
                  0.2040, 2.3951, 9.9726),     # panel 1
                 (0.0619, 2.1306, 10.0386, 0.5303, 3.4173, 10.4271,
                  1.1824, 4.4249, 11.0224),     # panel 2
                 (0.0592, 2.1138, 10.0372, 0.4950, 3.3478, 10.3825,
                  1.0752, 4.2732, 10.8964),     # panel 3
                 (0.0787, 2.1277, 10.0198, 0.6012, 3.4089, 10.2512,
                  1.2896, 4.4103, 10.6988)])    # panel 4

# Cox, Ingersoll, Ross (1985) Parameters
# from MS (2009), table 3, panel 1
kappa_r = 0.3
theta_r = 0.04
sigma_r = 0.1
r0 = 0.04

# Heston (1993) Parameters
# from MS (2009), table 3
para = array([(0.01, 1.5, 0.15, 0.1),     # panel 1
              (0.04, 0.75, 0.3, 0.1),     # panel 2
              (0.04, 1.5, 0.3, 0.1),     # panel 3
              (0.04, 1.5, 0.15, -0.5)])  # panel 4
theta_v = 0.02 # Long Term Volatility Level
S0 = 100.0     # Initial Index Level
```
### Numerical Results—5 simulation runs for the 36 American put options

<table>
<thead>
<tr>
<th>Overall Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start Calculations</strong></td>
</tr>
<tr>
<td><strong>Name of Simulation</strong></td>
</tr>
<tr>
<td><strong>Seed Value for RNG</strong></td>
</tr>
<tr>
<td><strong>Number of Runs</strong></td>
</tr>
<tr>
<td><strong>Time Steps</strong></td>
</tr>
<tr>
<td><strong>Paths</strong></td>
</tr>
<tr>
<td><strong>Control Variates</strong></td>
</tr>
<tr>
<td><strong>Moment Matching</strong></td>
</tr>
<tr>
<td><strong>Antithetic Paths</strong></td>
</tr>
<tr>
<td><strong>Option Prices</strong></td>
</tr>
<tr>
<td><strong>Absolute Tolerance</strong></td>
</tr>
<tr>
<td><strong>Relative Tolerance</strong></td>
</tr>
<tr>
<td><strong>Errors</strong></td>
</tr>
<tr>
<td><strong>Error Ratio</strong></td>
</tr>
<tr>
<td><strong>Aver Val Error</strong></td>
</tr>
<tr>
<td><strong>Aver Abs Val Error</strong></td>
</tr>
<tr>
<td><strong>Aver Rel Error</strong></td>
</tr>
<tr>
<td><strong>Aver Abs Rel Error</strong></td>
</tr>
<tr>
<td><strong>Time in Seconds</strong></td>
</tr>
<tr>
<td><strong>Time in Minutes</strong></td>
</tr>
<tr>
<td><strong>Time per Option</strong></td>
</tr>
<tr>
<td><strong>End Calculations</strong></td>
</tr>
</tbody>
</table>
Numerical Results—importance of algorithm features (I)

Table: Simulation results for different configurations of the LSM algorithm and an accuracy level of $PY_1 = 0.025$ and $PY_2 = 0.015$.

<table>
<thead>
<tr>
<th>$R$</th>
<th>$M$</th>
<th>$I$</th>
<th>CV</th>
<th>MM</th>
<th>AP</th>
<th>ATol</th>
<th>RTol</th>
<th>#Op</th>
<th>Err</th>
<th>AvEr</th>
<th>Sec/O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>15</td>
<td>25000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>1</td>
<td>-0.01</td>
<td>0.402</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>3</td>
<td>-0.012</td>
<td>0.54</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>25000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>1</td>
<td>-0.003</td>
<td>0.507</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>1</td>
<td>-0.012</td>
<td>0.699</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>25000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>2</td>
<td>-0.002</td>
<td>0.618</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>0</td>
<td>-0.008</td>
<td>0.894</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>0</td>
<td>-0.007</td>
<td>0.815</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>1</td>
<td>-0.007</td>
<td>1.012</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>35000</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>2</td>
<td>-0.006</td>
<td>0.67</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>35000</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>2</td>
<td>-0.006</td>
<td>0.963</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>35000</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>53</td>
<td>-0.019</td>
<td>0.737</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>35000</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>0.025</td>
<td>0.015</td>
<td>180</td>
<td>45</td>
<td>-0.013</td>
<td>1.262</td>
</tr>
</tbody>
</table>

$R =$ number of runs, $M =$ number of time intervals, $I =$ number of simulation paths, CV = control variates, MM = moment matching, $AP =$ antithetic paths, ATol = absolute performance yardstick, RTol = relative performance yardstick, #Op = number of options, Err = number of errors, AvErr = average error in currency units, Sec/O. = seconds per option valuation.
## Numerical Results—importance of algorithm features (II)

<table>
<thead>
<tr>
<th>$R$</th>
<th>$M$</th>
<th>$I$</th>
<th>CV</th>
<th>MM</th>
<th>AP</th>
<th>ATol</th>
<th>RTol</th>
<th>#Op</th>
<th>Err</th>
<th>AvEr</th>
<th>Sec/O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>20</td>
<td>35000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>13</td>
<td>-0.007</td>
<td>0.717</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>50000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>11</td>
<td>-0.008</td>
<td>1.507</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>75000</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>13</td>
<td>-0.011</td>
<td>2.096</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>75000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>10</td>
<td>-0.009</td>
<td>2.993</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>50000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>11</td>
<td>-0.007</td>
<td>3.231</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>75000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>7</td>
<td>-0.006</td>
<td>4.935</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>100000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>4</td>
<td>-0.007</td>
<td>6.044</td>
</tr>
<tr>
<td>5</td>
<td>75</td>
<td>100000</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.01</td>
<td>0.01</td>
<td>180</td>
<td>8</td>
<td>-0.006</td>
<td>8.696</td>
</tr>
</tbody>
</table>

Table: Simulation results for different configurations of the LSM algorithm and an accuracy level of $PY_1 = 0.01$ and $PY_2 = 0.01$.

Notation compare previous table.
Conclusions

1. a realistic market model should take into account stochastic volatility and short rates.
2. the valuation of non-vanilla options, like American put options, makes it necessary to use Monte Carlo simulation.
3. the simulation of complex models in combination with Least Squares Monte Carlo is computationally demanding and time consuming.
4. however, using a number of variance reduction techniques improves the performance (speed and accuracy) of the algorithm considerably.
5. Python is well-suited to implement efficient, i.e. fast and accurate, numerical valuation algorithms.
   - MCS with 25 steps/35,000 paths: 180 megabytes of data crunched in 0.9 seconds.
   - MCS with 50 steps/100,000 paths: 980 megabytes of data crunched in 6 seconds.
6. the speed-up compared to the times reported in MS2009 is 800+ times (0.9 seconds vs. 720+ seconds).
Contact

Dr. Yves J. Hilpisch
Visixion GmbH
Rathausstrasse 75-79
66333 Voelklingen
Germany
www.visixion.com
www.dexision.com
E contact@visixion.com
T +49 6898 932350
F +49 6898 932352
CV Yves Hilpisch

1. **1993–1996** Dipl.-Kfm. at Saarland University (Banks and Financial Markets)
2. **1996–2000** Dr.rer.pol. at Saarland University (Mathematical Finance)
4. **2005–present** Founder and MD of Visixion GmbH
   - Management and technical consulting work
   - Python programming (training, project work)
   - DEXISION—Derivatives Analytics On Demand (www.dexision.com)
5. **2010–present** Lecturer Saarland University
   - Lectures: “Numerical Methods for the Market-Based Valuation of Options”
   - Tutorials: Implementing financial models in Python
   - Book project: “Market-Based Valuation of Equity Derivatives—From Theory to Implementation in Python”
EXCURSION: Heston (1993) characteristic function

The characteristic function $\phi_{H}^{93}$ of the Heston stochastic volatility model is given by

$$\phi_{H}^{93}(u,T) = e^{H_{1}(u,T)+H_{2}(u,T)v_{0}}$$  \hspace{1cm} (10)$$

with the following definitions

$$c_{1} \equiv \kappa_{v}\theta_{v}$$
$$c_{2} \equiv -\sqrt{(\rho \sigma_{v}u_{i} - \kappa_{v})^{2} - \sigma_{v}^{2}(-u_{i} - u^{2})}$$
$$c_{3} \equiv \frac{\kappa_{v} - \rho \sigma_{v}u_{i} + c_{2}}{\kappa_{v} - \rho \sigma_{v}u_{i} - c_{2}}$$

$$H_{1}(u,T) \equiv r_{0,T}u_{i}T + \frac{c_{1}}{\sigma_{v}^{2}} \left\{ \kappa_{v} - \rho \sigma_{v}u_{i} + c_{2} \right\}T - 2 \log \left[ \frac{1 - c_{1}e^{c_{3}T}}{1 - c_{3}} \right]$$

$$H_{2}(u,T) \equiv \frac{\kappa_{v} - \rho \sigma_{v}u_{i} + c_{2}}{\sigma_{v}^{2}} \left[ \frac{1 - e^{c_{2}T}}{1 - c_{3}e^{c_{2}T}} \right]$$

and all variables as defined as in the main text. In $H_{1}$ we set $r_{0,T} = -\log(B_{0}(T))/T$. 
EXCURSION: Cox, Ingersoll, Ross (1985) bond pricing formula

The discount factor for discounting cash flows due at time $T$ to time $t = 0$, i.e. the present value of a zero-coupon bond paying one unit of currency at $T$, takes the form

$$B_0(T) = b_1(T)e^{-b_2(T)r_0}$$

(11)

$$b_1(T) \equiv \frac{2\gamma \exp(0.5(\kappa r + \gamma)T)}{2\gamma + (\kappa r + \gamma)(e^{\gamma T} - 1)} \frac{2\kappa_r \theta_r}{\sigma_r^2}$$

$$b_2(T) \equiv \frac{2\gamma}{2\gamma + (\kappa r + \gamma)(e^{\gamma T} - 1)}$$

$$\gamma \equiv \sqrt{\kappa_r^2 + 2\sigma_r^2}$$