"Vasile Alecsandri" University of Bacău
Faculty of Sciences
Scientific Studies and Research
Series Mathematics and Informatics
Vol. 19 (2009), No. 2, 479 - 492

PARALLEL ALGORITHMS FOR FINANCIAL DERIVATIVES EVALUATION IN GENERALIZED HESTON MODEL

TIBERIU SOCACIU, ILIE PARPUCEA, BAZIL PÂRV AND MARIA PÂRV

Abstract. This paper shows how can be estimated the value of an option if we assume the Heston model on a message-based architecture. We use two methods: first, a Monte Carlo method, then a parallelization of a recurrence obtained from a generalized Merton-Garman equation.

1. INTRODUCTION

From physical models, the following situation has reached acceptance: a financial asset *interest rate* follows a normal law, where the mean is the *drift rate* and the deviation is the *volatility*. This leads to a model that is currently accepted in finance:, the model of *geometric Brownian motion*. This model (known as *Black–Scholes–Merton model* in finance and financial engineering) is a stochastic differential equation:

$$dS(t) = mS(t)dt + sS(t)dB(t),$$

where:

a) $S(t), t \geq 0$ is a stochastic process for the *value* of stock;

b) m is a static parameter for the *drift rate of return*;

c) s^2 is a static parameter for the *volatility of stock* ($s \geq 0$);

d) $B(t), t \geq 0$ is a *standard Wiener process*.

Key words and phrases: parallel algorithms; computational financial engineering; derivatives evaluation; Black–Scholes–Merton model; generalized Heston model; Black–Scholes equation; generalized Merton–Garman equation.

(2000) Mathematics Subject Classification: 35Q80, 62M05, 65Y05, 65Y10, 68W15, 91B70.

Another model is assumed by Heston (see [1]) and it consists from two stochastic differential equations: The Heston model corrects some inconsistency of the Black–Scholes–Merton model, for example:

a) in reality, volatility is not a static parameter; it can be used as static value only on short periods (this value will obtain on calibration process, usual with a *statistical estimator*);

b) on long periods, it is possible that interest rate series did not verify a normal law.

A generalization of Heston model is described by the following coupled stochastic differential equations:

$$dS(t) = A(S(t), v(t), t)dt + B(S(t), v(t), t)dB_1(t)$$
$$dv(t) = C(S(t), v(t), t)dt + D(S(t), v(t), t)dB_2(t)$$

where:

a) $S(t), t \geq 0$ is a stochastic process for value of stock;

b) $v(t), t \geq 0$ is a stochastic process for volatility of value of stock;

c) $A(S, v, t), B(S, v, t), C(S, v, t), D(S, v, t)$ are three parametric algebraic functions;

d) $B_1(t), B_2(t)$, $t \geq 0$ are two *r correlated* standard Wiener processes, i.e.:

$$dB_1(t)dB_2(t) = rdt$$

For Wiener processes, more details can be found in [2].

For the basic Heston model we have:

a) $A = S(t)m$

b) $B = S(t)\sqrt{v(t)}$

c) $C = K\left(\theta - v(t)\right)$

d) $D = \xi\sqrt{v(t)}$

where:

a) m is a drift of rate;

b) θ is *long run average price volatility*; as t tends to infinity, the expected value of *v(t) tends* to θ;

c) K is the rate at which *v(t) reverts* to θ;

d) ξ is the *volatility of the volatility*; as the name suggests, this determines the variance of *v(t)*.

Note that for $C = D = 0$ we obtain a *static volatility* model (Black–Scholes–Merton):

$$dv(t) = 0 .$$

Any financial derivative based on support with price *S(t)* at time *t,* with quotation at time *t* and a value *S* of support as *V(t,S)*, where $V : [0,T] \times R_+ \to R_+$ and at maturity time T will generate an generate an interest rate: $payoff : R_+ \to R_+$

For example, European options CALL and PUT has payoff functions:

$$payoff(\mathrm{x}) = \max\{0, \mathrm{x} - \mathrm{E}\}$$
$$payoff(\mathrm{x}) = \max\{0, \mathrm{E} - \mathrm{x}\}$$

2. PARALLELIZATION OF A MONTE CARLO METHOD FOR OPTION'S EVALUATION

First, we discretize continuous dimension of time. Let us denote:

$$t[k] = t[0] + k\Delta,\ 0 \le \mathrm{k} \le \mathrm{N},\ \Delta = \frac{T - t[0]}{N}$$

where:

a) *T* is the maturity time of option;

b) *N* is a number of time units (like days); note that sometimes is used transaction days – in this case, discretization hasn't a constant step.

Because for a standard Wiener process *B(t), t ≥ 0* we can obtain a standard normal random variable series *X[B(t)], t ≥ 0* with:

$$dB(t) = X\sqrt{dt}$$

we can build a simulation step as:

A = A(S[k], v[k], t[k])
B = B(S[k], v[k], t[k])
C = C(S[k], v[k], t[k])
D = D(S[k], v[k], t[k])
S[k+1] = S[k] + A Δ + B X SQRT(Δ)
v[k+1] = v[k] + C Δ + D Y SQRT(Δ)

where X and Y are r correlated. A simple method to generate two r correlated normal values is:

X = NORMRAND()
Z = NORMRAND()
$Y = r\,X + SQRT(1–r^2)\,Z$

A complete simulation for interval [t0, T] in N step with evaluation of payoff is:

```
FUNCTION simulation()
S = S0
v = v0
t = t0
Δ = (T – t0) / N
FOR k = 1, N
    t = t + Δ
    X = NORMRAND()
    Z = NORMRAND()
    Y = r X + SQRT(1–r^2) Z
    A = A(S, v, t)
    B = B(S, v, t)
    C = C(S, v, t)
    D = D(S, v, t)
    SS = S + A Δ + B X SQRT(Δ)
    vv = v + C Δ + D Y SQRT(Δ)
    S = SS
    v = vv
END FOR
RETURN S
END FUNCTION
```

Because for a level of acceptance α, where $0<\alpha<1$, a trust interval for E[S(T)] is [s – a, s + a], where

$$s = \sum_{1\le k\le M} simulation()$$

$$a = \frac{F\left(\frac{\alpha}{2}\right)\sigma}{\sqrt{M}},$$

and F is the inverse function for CDF (cumulative distribution function) of standard normal distribution; it means that:

$$Prob(s - a < E[S(T)] < s + a) = 1 - a$$

or:

$$\text{Prob}(E[S(T)] = s + O(\sqrt{M})) = 1 - a .$$

where *big–O* notation is a *Buchmann–Landau symbol* (see [3]).

Suppose that we have a parallel architecture with message passing protocol (like MPI; for MPI see [5]) that consists in P processors. We broke a simulation job for P–1 slaves which will transmit result of $\left\lceil \frac{M}{P-1} \right\rceil$ simulations to master processor:

```
PROGRAM complete_simulation
GLOBAL t0, T, S0, v0, P, N, EPSILON, M
READ t0, T, S0, v0, N, EPSILON, M
P = ProcessorsCount()
If P ≥ 3 THEN
      CALL parallel_simulation()
ELSE
     CALL serial_simulation()
END IF
END PROGRAM

PROCEDURE parallel_simulation()
LOCAL x, y, Q
IF ProcessorID() = 0 THEN
     FOR i = 1, P–1
        RECV ANY, y
        x = x + y
     ENDFOR
     x = payoff(x / (P–1))
ELSE
     Q = CEIL(M / (P–1))
     x = 0
     FOR i = 1, Q
        x = x + simulation()
     ENDFOR
     SEND 0, x / Q
ENDIF
END PROCEDURE
```

```
PROCEDURE serial_simulation()
LOCAL x
x = 0
FOR i=1, M
   x = x + simulation()
ENDFOR
WRITE payoff(x / M)
END PROCEDURE
```

3. PARALLELIZATION OF NUMERICAL METHOD APPLIED TO GENERALIZED MERTON–GARMAN EQUATION

We recall the following extended Itô's Lemma. For Itô's lemma see [4].

Theorem. Let S(t), t≥0 and v(t), t≥0 be two stochastic processes that verify the next differential stochastic equations:

$$dS(t) = A(S(t), v(t), t)dt + B(S(t), v(t), t)dB_1(t)$$
$$dv(t) = B(S(t), v(t), t)dt + D(S(t), v(t), t)dB_2(t)$$

where $B_1(t), t = 0$ and $B_2(t), t = 0$ are two Wiener correlated processes with ρ corellation. If f(S, v, t) is a continuous differentiable function, then:

$$df = \left[f_t + f_S A + f_v C + \frac{f_{SS} B_2}{2} + \frac{f_{vv} D^2}{2} + f S_v \rho BD \right] dt + f_S B dB_1 + f_v D dB_2$$

Proof: see [6]

Because we have two sources of risk, we must build a portfolio based on two types of derivatives and support:

a) one option at value $V(t, S, \sigma)$;

b) X shares at price $S(t)$;

c) Y options at value $V_1(t, S, \sigma)$.

Value of this portofolio is:

$$\Pi(t) = V(t, S, \sigma) + XS(t) + YV_1(t, S, \sigma)$$

After appling extended Itô's lemma for:

$$f_1(S,\sigma,t) = V_1(t,S,\sigma)$$
$$f_2(S,\sigma,t) = V\ (t,S,\sigma)$$

we can compute:

$$d\Pi(t) = dV(t,S,\sigma) + X\,dS(t) + Y\,dV_1(t,S,\sigma)$$

as:

$$d\Pi(t) = \alpha_1 dt + \alpha_2\, dB_1(t) + \alpha_3\, dB_2(t)$$

Because is a risk–free portofolio, we must have:

$$\alpha_2 = \alpha_3 = 0\,.$$

But for a risk–free portofolio we have:

$$d\Pi(t) = r\Pi(t)\,dt\,.$$

From last two we obtain that (see [6]):

$$\frac{\mathrm{rV} + \mathrm{V_s rS} + \mathrm{V_t} + \mathrm{AV_S} + \mathrm{CV_v} + \frac{\mathrm{B^2 V_{SS}}}{2} + \frac{\mathrm{D^2\ V_{vv}}}{2} + \rho\mathrm{BDVS_v} - \mathrm{AV_S}}{DV_v},$$

is an invariant denoted as β, and named as *the market price of volatility risk* (see [7], p8). Rewrite last as a partial differential equation:

$$\mathrm{V_t} + \mathrm{rSV_S} + (\mathrm{C} - \mathrm{D}\beta)\mathrm{V_v} + \frac{\mathrm{B^2 V_{SS}}}{2} + \frac{\mathrm{D^2 V_{vv}}}{2} + \rho\mathrm{DVS_v} - \mathrm{rV} = 0$$

obtain an equation like Black–Scholes equation (more on Black–Scholes equation in [8]). For Heston model, this equation is named as *Merton–Garman equation* (see [9]; [10] p. 41) or *Garman equation*. We name as *generalized Merton–Garman equation* or *generalized Garman equation*.

On boundary we have:

a) value of a null valued stock is null:

$$V(S,v,T) = 0, \forall \mathrm{v}, \forall \mathrm{t}, 0 \le \mathrm{t} \le \mathrm{T}\,,$$

b) value at maturity will be payoff():

$$V(S,v,T) = payoff(S), \forall v, \forall S, S \ge 0\,.$$

c) for null volatility we have a deterministic solution S^* of differential equation:

$$dS(t) = A(S(t), 0, t)\, dt$$

and value of option is:

$$V(S,0,t) = S^*(t), \forall S, \forall t, S \geq 0, 0 \leq t \leq T .$$

In particular case of Heston model:

$$A = r\, S,$$

solution S^* is:

$$S^*(t) = S(t0) \exp (r\, (t - t0))$$

After discretization on all three axis with ΔS, Δv, Δt denote with:

$$V[i,j,k] = V(j(\Delta S), k(\Delta v), i(\Delta t))$$

generalized Merton–Garman equation can be rewritten as:

$$\frac{V[i,j,k]\ V[i-1,j,k]}{\Delta t} + \frac{rj\,(\Delta S)\,(V[i,j+1,k] - V[i,j1,k])}{2\Delta S} +$$
$$+ \frac{(C - D\beta)\,(V[i,j,k1] - V[i,j,k-1])}{2\Delta v} + \frac{B^2\,(V[i,j+1,k] - 2\,V[i,j,k] + V[i,j-1,k])}{2(\Delta S)^2} +$$
$$+ \frac{D^2\,(V[i,j,k+1] - 2\,V[i,j,k] + V[i,j,k-1])}{2(\Delta v)^2} +$$
$$+ \frac{\rho BD\,(V[i,j+1,k+1] - V[i,j+1,k-1] - V[i,j-1,k+1] + V[i,j-1,k-1])}{(\Delta S)(\Delta v)} -$$
$$- rV[i,j,k] = 0$$

or in liniar form explicit form:

$$V[i-1,j,k] = p\, V[i,j,k] + q\, V[i,j+1,k] + r\, V[i,j-1,k] +$$
$$+ s\, V[i,j,k+1] + t\, V[i,j,k-1] + u\, V[i,j+1,k+1] -$$
$$- u\, V[i,j+1,k-1] - u\, V[i,j-1,k+1] + u\, V[i,j-1,k-1]$$
$$-$$

where:

$p = (a - 2\,d - 2\,e - g) / a$
$q = (d + b) / a$
$r = (d - b) / a$
$s = (e + c) / a$
$t = (e - c) / a$

$u = f / a$
$a = (\Delta S)^2 (\Delta v)^2$
$b = ½ r j (\Delta t) (\Delta S)^2 (\Delta v)^2$
$c = ½ (C – D \beta) (\Delta t) (\Delta S)^2 (\Delta v)$
$d = ½ B^2 (\Delta t) (\Delta v)^2$
$e = ½ D^2 (\Delta t) (\Delta S)^2$
$f = \rho B D (\Delta t) (\Delta v) (\Delta S)$
$g = r (\Delta t) (\Delta v)^2 (\Delta S)^2$

A parallel solution for a PRAM (see [11]) architecture was presented in [12]. We will build a parallel solution for a message passed architecture with a diagonalization method of domain splitting in order to reduce the number of messages (like in [13], where communication was halfed). Supose that our architecture has Q processors, where one will be master and Q-1 will be slaves.

Each slave will process a prism obtained from cube

$$\text{CUBE} = [0, M] \times [0, N] \times [0, P],$$

where M is the number of epochs (time), N–1 is number of intermediar points on S axis, with N (ΔS) enough biger to cover values, respectively P–1 for v axis, after split it in Q-1 prisms:

$$\text{PRISM}[x] = [0, M\} \times \{(y, z) : f(x-1) \le y + z \le f(x)\}$$

where:

$$f(x) = \text{FLOOR}\ (\ (N + P + 1)\ x\ /\ (Q - 1)\ ).$$

Because processing of PRISM[x] need first 2 left layer from PRISM[x+1] and last 2 right layer from PRISM[x-1], processors must interchange some values M-1 times. Parallel algorithm is:

```
PROGRAM simulation
Q = ProcessorsCount()
x =  ProcessorID()
IF x = 0 THEN
  Z = 0
  WHILE Z < Q – 1 DO
    RECV x, I, J, K, a
    IF I = J = K = 0 THEN
      Z = Z + 1
```

```
   ELSE
    V[I, J, K] = a
   ENDIF
  END WHILE
ELSE
  FOR j = 0, N
   FOR k = 0, P
    V[M, j, k] = payoff(j (ΔS))
   ENDFOR
  ENDFOR
  FOR I = M, 1, -1
   // border
   FOR k = 0, P
    V[i-1, 0, k] = 0
   ENDFOR
   FOR j = 0, N
    V[i-1, j, 0] = S*((i-1) (Δt))
   ENDFOR
   // first 2 layers
   If x <> 1 THEN
    RECV x-1, {V[i,j,k]} | j+k=f(x-1)-1 OR  j+k=f(x-1)-2}
   ENDIF
   FOREACH j,k WITH j+k=f(x-1) OR  j+k=f(x-1)+1
    CALL compute_one()
   ENDFOR
   If x <> 1 THEN
    SEND x-1, {V[i,j,k]} | j+k=f(x-1) OR  j+k=f(x-1)+1}
   ENDIF
   // other internal layers
   FOREACH j,k WITH f(x-1)+2<=j+k<=f(x)-2
    CALL compute_one()
   ENDFOR
   // last 2 layers
   If x <> Q-1 THEN
    RECV x+1, {V[i,j,k]} | j+k=f(x)+1 OR  j+k=f(x)+2}
   ENDIF
   FOREACH j,k WITH  j+k=f(x) OR  j+k=f(x)-1
    CALL compute_one()
   ENDFOR
```

```
   If x <> Q-1 THEN
     SEND x+1, {V[i,j,k]} | j+k=f(x) OR  j+k=f(x)-1}
   ENDIF
  ENDFOR
  SEND 0, 0, 0, 0, 0
ENDIF
END

PROCEDURE compute_one()
IF j = 0 OR k = 0 THEN
 RETURN
ENDIF
A = A(j (ΔS), k (Δv), i (Δt))
B = B(j (ΔS), k (Δv), i (Δt))
C = C(j (ΔS), k (Δv), i (Δt))
D = D(j (ΔS), k (Δv), i (Δt))
a = (ΔS)^2 (Δv)^2
b = ½ r j (Δt) (ΔS)^2 (Δv)^2
c = ½ (C – D β) (Δt) (ΔS)^2 (Δv)
d = ½ B^2 (Δt) (Δv)^2
e = ½ D^2 (Δt) (ΔS)^2
f = ρ B D (Δt) (Δv) (ΔS)
g = r (Δt) (Δv)^2 (ΔS)^2
p = (a – 2 d – 2 e – g) / a
q = (d + b) / a
r = (d – b) / a
s = (e + c) / a
t = (e – c) / a
u = f / a
V[i-1,j,k] = p V[i,j,k] + q V[i,j+1,k] + r V[i,j–1,k]
  + s V[i,j,k+1] + t V[i,j,k–1] + u V[i,j+1,k+1]
  – u V[i,j+1,k–1] – u V[i,j–1,k+1] + u V[i,j–1,k–1]
SEND 0, i-1, j, k, V[i-1,j,k]
END
```

Acknowledgment. This work was supported by the grant ID 2585, sponsored by NURC – Romanian National University Research Council (CNCSIS – Consiliul National al Cercetarii Stiintifice in Invatamantul superior, see [14]).

References

[1] Steven L. Heston, **A closed–form solution for options with stochastic volatility with applications to bond and currency options**, in The Review of Financial Studies, 1993, Volume 6, number 2, pp. 327–343.

[2] Daniel Revuz, Marc Yor, **Continuous martingales and Brownian motion**, second edition, Springer–Verlag 1994.

[3] Eric W. Weisstein, **Landau Symbols**, in MathWorld—A Wolfram Web Resource, online at http://mathworld.wolfram.com/LandauSymbols.html, last access: 13 jun. 2009.

[4] Kiyoshi Itō, **On stochastic differential equations**, in Memoirs of American Mathematical Society , 1951, 4, pp. 1–51.

[5] John M. McQuillan, David C. Walden, **Some considerations for a high performance message-based interprocess communication system**, in Proceedings of the 1975 ACM SIGCOMM/SIGOPS workshop on Interprocess communications, ACM Press, 1975.

[6] Tiberiu Socaciu, **Obtaining Generalized Garman Equation**, in Dinu Airinei et al (eds), "Globalization and Higher Education in Economics and Business Administration", Tehnopress, Iasi 2009, pp. 53-58.

[7] Nimalin Moodley, **The Heston Model: A Practical Approach with Matlab Code**, bachelor thesis, University of the Witwatersrand, Johannesburg, South Africa, online at http://www.math.nyu.edu/~atm262/fall06/compmethods/a1/nimalinmoodley.pdf, last access: 1 may 2009.

[8] Fischer Black, Myron Scholes, **The Pricing of Options and Corporate Liabilities**, in Journal of Political Economy, 81 (3), pp. 637–654.

[9] Marakani Srikant, **Option Pricing with Stochastic Volatility**, Department of Computational Science, National University of Singapore, 1997/1998, online at http://srikant.org/thesis/thesis.html, last access: 1 may 2009.

[10] Belal E. Baaquie, **Quantum Finance. Path Integrals and Hamiltonians for Options and Interes Rates**, Cambridge Press, 2004,

332 pages, ISBN 9780521840453, partial online on books.google.com, last access: 1 may 2009.

[11] Xingzhi Wen, Uzi Vishkin, **PRAM-On-Chip: First Commitment to Silicon**, online at http://www.umiacs.umd.edu/users/vishkin/XMT/spaa07paper.pdf, last access at 1 august 2009.

[12] Tiberiu Socaciu, Ioan Maxim, Mirela Danubianu, **Parallel numerical algorithm for Black–Scholes like PDE from Heston's model**, at A XXV-a Conferință Națională Didactica Matematicii, Turda, 23 mai 2009 (Algoritmi paraleli pentru rezolvarea numerica a ecuatiei de tip Black-Scholes de la modelul lui Heston, http://www.math.ubbcluj.ro/~didactica/).

[13] Tiberiu Socaciu, Mirela Danubianu, **Two Parallel Solutions of Black–Scholes Equation**, in Journal of Applied Computer Science and Mathematics, 5 (2009), online at http://jacs.usv.ro/getpdf.php ? issue=5&paperid=55, last access 1 august 2009..

[14] http://www.cncsis.ro/PNCDI%20II/Idei/2008/Exploratorie/Prop_finantare/Proiecte_propuse_spre_finantare_domeniul_13_3B.html last access 1 august 2009.

Tiberiu Socaciu
Faculty of Economics and Public Administration,
Department of Informatics
"Ştefan cel Mare" University of Suceava,
Suceava, Romania
and
Faculty of Informatics
Vasile Goldis West University of Arad,
Arad, Romania
socaciu@seap.usv.ro

Ilie Parpucea
Faculty of Economics and Affairs,
Department of Statistics–Forecasts–Mathematics,
"Babeş–Bolyai" University of Cluj–Napoca
Cluj–Napoca, Romania
parpucea@econ.ubbcluj.ro

Bazil Pârv
Faculty of Mathematics and Informatics,
Department of Programming Languages and Methods
"Babeş–Bolyai"University of Cluj–Napoca
Cluj–Napoca, Romania, bparv@cs.ubbcluj.ro

Maria Pârv
Department of Mathematics–Informatics
University of Agricultural Sciences and Veterinary Medicine of Cluj–Napoca
Cluj–Napoca, Romania
maria_parv@yahoo.com