"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

PARALLELIZING BLACK-SCHOLES PDE SOLVING ON AN OPEN DOMAIN WITH SCHUR DECOMPOSITION METHOD

Tiberiu SOCACIU*, Ioan MAXIM***

*Faculty of Economics, "Stefan cel Mare" University of Suceava, Romania
** Faculty of Informatics, "Vasile Goldis" West University of Arad, Romania,
*** Department of Teacher Training, "Stefan cel Mare" University of Suceava, Romania

Abstract: *In this paper we will provide some parallel algorithms for PRAM and BSP architectures for solving Black-Scholes PDE on an open domain (on support, non temporal dimension) and without frontier condition there. We suppose exist an PDE solver (like an implicit method based on algebraic system solver) for a rectangle with conditions only on two adiacent lines from border, after computation we will use computed value for an other rectangle with same PDE solver.*

Mathematics Subject Classifications 2010: *65C99, 65N06, 65Y05.*
Keywords: *parallelization, Black-Scholes PDE, Schur decomposition method.*

1. BLACK-SCHOLES-MERTON MODEL AND BLACK-SCHOLES PDE

One of popular stochastic equation that models a traded asset (like a stock) is the *Black-Scholes-Merton model* based on *geometric brownian motion* (see [1]):

$$dS(t)=S(t)[\mu dt+\sigma dW(t)] \quad (1)$$

where $(S(t), t\geq 0)$ is the stochastic process for asset value at timestamp t, $(W(t), t\geq 0)$ is a *Wiener standard process* (see [2]), μ is the *drift rate of return* and σ is *volatility*.

A derivative based on this asset is an other traded asset that fructify at *maturity time* T, payoff depend on value of support, S(T). *payoff* function is defined as:

$$payoff:R_+\rightarrow R \quad (2)$$

Main problem is pricing of a financial derivative. For this, we build an risk-free portofolio based on some supports and some derivatives. After applying of Ito lemma (see [3]) we obtain Black-Scholes PDE (see [4]):

$$V_t+½\sigma^2S^2V_{SS}+rSV_S-rV = 0 \quad (3)$$

for pricing derivatives, where:

$$V:R_+x[0,T]\rightarrow R_+ \quad (4)$$

and V(S,t) is value of derivatives at timestamp t if support is valued as S.

Note that for a generalized brownian motion:

$$dS(t)=A(S(t),t)dt+B(S(t),t)dW(t) \quad (5)$$

where A(S,t) and B(S,t) are some algebraic expression, we can build a generalized form of Black-Scholes PDE (see [5]):

$$V_t+\frac{1}{2}B^2V_{SS}+rSV_S-rV = 0 \qquad (6)$$

Black-Scholes PDE and generalized Black-Scholes PDE can be linked with Dirichlet condition:

$$V(0,t)=0 \qquad (7)$$
$$V(S,T)=\text{payoff}(S) \qquad (8)$$

that means for 0 value of support, derivatives is valued to 0 too, and value at maturity is payoff function.

2. SCHUR DECOMPOSITION METHOD FOR BLACK-SCHOLES PDE ON AN OPEN DOMAIN

Schur method (see [6]) is based on decomposition of an initial domain D of a Dirichlet problem Lu=0 in two or more problems Lu=0 defined on domains D_1, D_2, …, where union of domains cover D:

$$D \subseteq D_1 \cup D_2 \cup \ldots \qquad (9)$$

with Dirichlet conditions on some frontiers as subset of $\partial D_i \cap \partial D$

The main idea to solve generalized Black-Scholes PDE (6) is splitting domain D in rectangles ($D_{i,j}$, i=1,N, j≥1)

$$D_{i,j} = \{ (s, t) \mid t\in[(N+1-i)T/N, (N-i)T/N], s\in[(j-1)\Delta, j\Delta] \} \qquad (10)$$

like in figure 1:

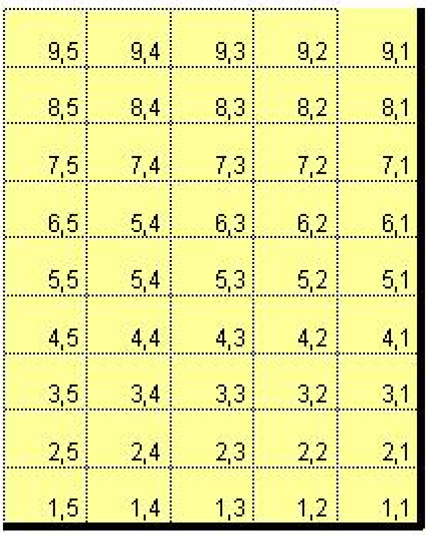

Figure 1. Decompozition of D (partial) in $D_{i,j}$

and solving PDE successively on $D_{1,1}$, $D_{1,2}$, $D_{2,1}$, $D_{1,3}$, $D_{2,2}$, $D_{3,1}$, etc like in figure 2.

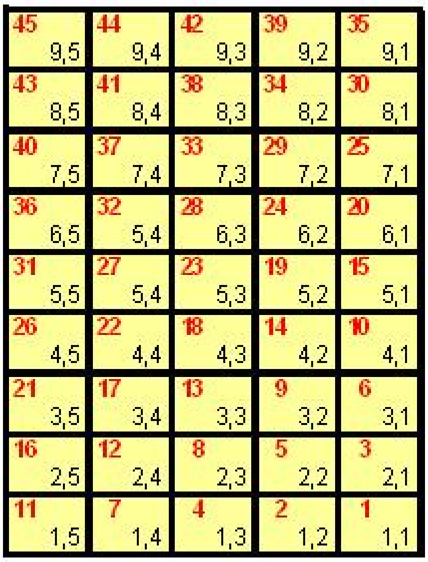

Figure 2. Succesion on solving PDE on $D_{i,j}$

If suppose that SOLVERECTANGLE(i,j) solve Black-Scholes PDE on $D_{i,j}$ with any method (like an implicit method) in Dirichlet conditions un right and bottom frontiers like in figure 3:

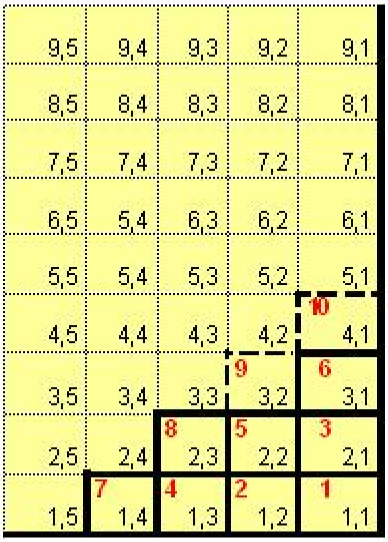

Figure 3. Right and bottom Dirichlet condition is precomputed on previous call of SOLVERECTANGLE(i,j)

Serial algorithm for this method is:

Algorithm SchurSerial
For K=2 to M+N
  For I=max(1,K-N) To min(M,K-1)
    SOLVERECTANGLE(I,K-I)
  End For
End For
End Algorithm

3. PARALLELIZATION OF SCHUR METHOD FOR A PRAM ARCHITECTURE

A good idea for parallelization of previous algorithm on a PRAM architecture (see [7]) is parallel execution of internal FOR that means all $D_{i,j}$ on same diagonal (i+j invariant) can be processed in parallel. Parallel PRAM Algorithm is:

Algorithm SchurPRAM
For K=2 to M+N
  // diagonal can be processed
  // in parallel
  Parallel For I=max(1,K-N) To min(M,K-1)
    SOLVERECTANGLE(I,K-I)
  End For
End For
End Algorithm

In figure 4 is shown with same colour parallel PRAM steps:

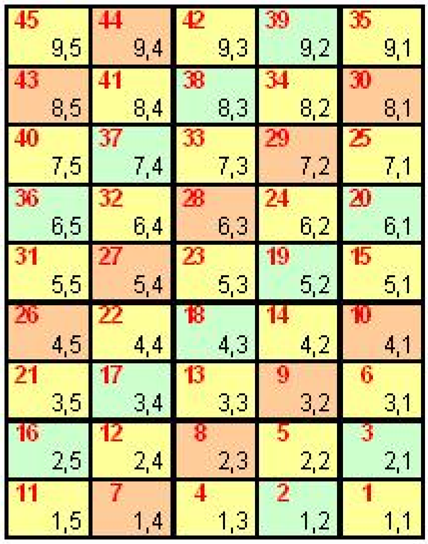

Figure 4. Coloured parallel PRAM steps

Main problem is when min(M,N) mod P is not null, that means some processors will be free at last part of parallel for execution. Is a good ideea to try min(M,N) as a multiply of P, because all calls of *SOLVERECTANGLE* are similar, because problems are similar on each subdomains.

4. PARALLELIZATION OF SCHUR METHOD FOR A BSP ARCHITECTURE

For a BSP architecture (see [8]) that has P=min(M,N) processors, that can process in parallel an entire diagonal of subdomains (i+j = K).

In figure 5, domains coloured with same colour will be processed as same processor:

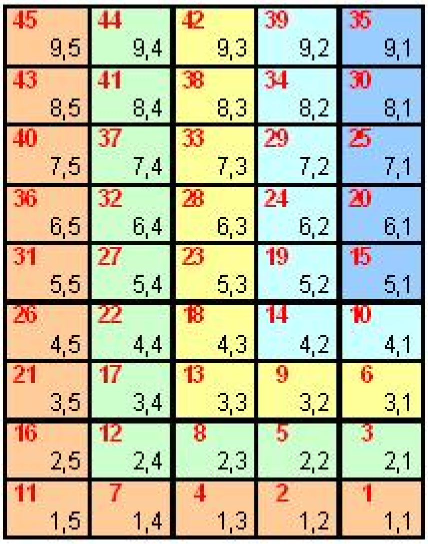

Figure 5. Coloured BSP processor assign

Parallel algorithm for BSP architecture is:

```
Algorithm SchurBSP
For K=2 to M+N
  Macrostep Start
  A= max(1,K-N)
  B= min(M,K-1)
  For I=A To B
    If ProcessorId = I-A+1 Then
      Call SOLVERECTANGLE(I,K-I)
    End If
    // sincronize border data with other
    // processors
    If K<min(M,N) then
      If ProcessorId < min(M,N) then
        SendBorderData(ProcessorId+1)
      End If
    Else
      If ProcessorId > 1 Then
        SendBorderData(ProcessorId-1)
      End If
    End If
  End For
  MacroStep Stop
End For
End Algorithm
```

REFERENCES

1. Pavel Fărcaş, Radu Moleriu, *Elemente de probabilităţi şi teoria proceselor stochastice cu aplicaţii în matematica financiară*, Editura Albastra, Cluj-Napoca, 2006.
2. Daniel Revuz, Marc Yor, *Continuous martingales and Brownian motion*, second edition, Springer-Verlag 1994.
3. Kiyoshi Itō, *On stochastic differential equations*, in *Memoirs*, American Mathematical Society, 1951, 4, pp. 1–51.
4. Fischer Black, Myron Scholes, *The Pricing of Options and Corporate Liabilities*, in *Journal of Political Economy*, 1973, 81 (3), pp. 637–654.
5. Tiberiu Socaciu, *Metode numerice de evaluare a optiunilor financiare*, Editura InfoData, Cluj 2010.
6. De Alfio Quarteroni,Alberto Valli, *Domain decomposition methods for partial differential equations*, Oxford University Press, 1999, partial online at http://www.google.com/books?hl=ro&lr=&id=CswwZ9Ca9IIC&oi=fnd&pg=PA1&dq=schur+decomposition+method&ots=emJ2evEUra&sig=EnVc2pTCOULgVZ4Y8V7suN2vB5k#v=onepage&q=schur%20decomposition%20method&f=false, last access at 2010 apr. 6.
7. Dan Grigoras, *Calcul paralel. De la sisteme la programarea aplicaţiilor*, Computer Press Agora, Cluj-Napoca, 2000.
8. D.B. Skillicorn, Jonathan M.D. Hill, W.F. McColl, *Questions and answers about BSP*, Oxford University Computing Laboratory, 11 nov. 1996, online at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.9325, last access at 2010 apr 6.