 CMES, vol.86, no.2, pp.93-112, 2012

Simulation of Multi-Option Pricing on Distributed Computing

J.E. Lee[1] **and S.J. Kim**[2]

Abstract: As the option trading nowadays has become popular, it is important to simulate efficiently large amounts of option pricings. The purpose of this paper is to show valuations of large amount of options, using network distribute computing resources. We valuated 108 options simultaneously on the self-made cluster computer system which is very inexpensive, compared to the supercomputer or the GPU adopting system. For the numerical valuations of options, we developed the option pricing software to solve the Black-Scholes partial differential equation by the finite element method. This yielded accurate values of options and the Greeks with reasonable computational times. This was executed on the single node and then extended on the cluster computer system.

We can infer our research for large amounts options on the distributed computing will be a highly attractive alternative to devising hedging strategies or developing new pricing models.

Keywords: option pricing model, Black-Scholes equation, finite element method, distributed computing, cluster computer system.

1 Introduction

Since the introduction of the option pricing model by Black and Scholes (1973), various option products based on the Black-Scholes equation have been developed. In general, option products preclude a closed-form solution except for the well-known classical cases, such as the European options. When a closed-form solution does not exist, numerical approaches are common way to proceed. Over the past years, several numerical methods have been introduced and applied to the option pricing models. Generally, binomial tree method and finite difference method are

[1] Interdisciplinary program in computational science & Technology, Seoul National University, Seoul, Korea

[2] Department of Aerospace engineering, Seoul National University, Seoul, Korea. Corresponding Author: J.E. Lee, E-mail: jelee@aeroguy.snu.ac.kr

in favor, as they are intuitive and easy; the Monte Carlo simulation is also popular. More recently, the finite element method (FEM), which is widely used in others fields of science and engineering for decades, is becoming popular in financial engineering applications, by the number of articles being published on the subject, such as Topper (2005a, 2005b), Achdou and Pironneau (2005), Seydel (2009), Tomas III and Yalamanchili (2001), Sapariuca, Marcozzib and Flahertyc (2004) and Foufas and Larson (2008).

The FEM is advantageous over other numerical methods when properly applied to the problems of financial modeling. For example, Seydel (2009; chap 5) illustrated triangular elements for the double barrier option to show the flexibility of domain discretization, which is offered by the FEM. Tomas and Yalamanchili (2001) presented the advantages of the FEM, showing non-uniform mesh construction and direct derivative valuation. Various option products applied FEM can be found in Topper (2005a). Topper also pointed out the FEM can incorporate different kinds of boundary conditions in an easy way and can easily deal with high curvature and irregular shapes of the computational domain, compared to other numerical techniques. These are important advantages, in practice, in the field of financial modeling. On the other hand, various new techniques have been studied to improve the computational efficiency. For example, Jackson and Süli (1997) introduced an adaptive mesh refinement technique for the option pricing, based on a posteriori error estimation, and Foufas and Larson (2008) developed it, calculating optimal meshes for each type of option. More information about the mesh adaption can be found in Achdou and Pironneau (2005; chap 5). Zvan, Forsyth and Vetzal (1998) proposed the penalty method by adding a penalty term to the Black-Scholes equation, and Nielsen, Skavhaug and Tveito (2002, 2008) extended and refined it. Khaliq, Voss and Kazmi (2006, 2008) developed an efficient implicit scheme using the penalty method. Other techniques are the MQ-RBF (Choi and Marcozzi (2004)), the LDQ method (Young, Sun and Shen (2009)), PSM (Suh (2009)), the spectral element method (Zhu and Kopriva (2010)), STS technique (O'sullivan and O'sullivan (2011)), etc. Some articles compared existing numerical schemes, conducting a computational study based on the speed of computation and the accuracy of the result (Broadie and Detemple (1996), Rogers and Talay (1997), Cooney (2000) and Wallner and Wystup (2004)).

We used the FEM to solve numerically the Black-Scholes partial difference equation for the option pricing. Our approach is general, practical and easily implemented for the numerical modeling, and the computation performance is very positive. This means that our approach is very easy to use and convenient in real simulations.

Meanwhile, a lot of financial companies are handling various types of complex financial products using high performance computing, accelerated by GPGPU (General-Purpose computing on Graphics Processing Units) technique or supercomputing systems. They valuate worldwide market option products or compute hedge strategies for hundreds of thousands of policy holders in its portfolio. The finance is one of the fast growth fields for high performance computing systems, driven by increasing data volumes, greater data complexity.

In this paper, instead of building a supercomputer or installing more powerful GPUs, which is expensive and sometimes not possible, we suggest the cluster computer system by connecting many low-cost standard computers. This system is easy to improve the performance by adding nodes and can use the out of date computers. This could be the most economical way to take the high performance computing.

The main contributions and findings in this paper are following.

We developed the option pricing software applying the FEM. This program provides option prices and the corresponding Greeks. This is supporting both one-underlying asset option and two-underlying assets option, and the computation performance is very positive as the desired level of accuracy is achieved.

About one hundred options' valuations were performed simultaneously on the cluster computer system developed by our lab, with a CPU time of a few seconds. Our results could be very useful to study how option pricings are changing with various conditions, which will be able to take great promise for improving the finance modeling and analysis

Option pricing on the distributed computing is for the first time in the financial fields, to our knowledge, and this is a very economical way, compared to other high performance systems such as the supercomputer or the GPU adopting system.

The remainder of this paper is organized as follows. Section 2 presents the option pricing model defined by a partial differential equation and its boundary conditions. This model is based on the Black-Scholes equation. Section 3 introduces the FEM for the option pricing numerical model and Section 4 shows the development of the option pricing program based on the FEM. Section 5 describes the cluster computer system we used. Section 6 presents the numerical results on the single node and 108 nodes in the cluster system, which is the main contribution of this paper. Finally, conclusions are made in Section 7.

2 The option pricing model

Option is the right to buy or sell an asset. This gives its holder the right for a transaction of a certain asset at a given time for a given price. The holder does not have to exercise this right. This fact distinguishes options from forwards and futures where the holder is obligated to buy or sell the underlying asset. There are two types of options. A call option allows its owner to buy and a put option to sell its underlying asset at a certain time for a fixed certain price. The price is known as the exercise price or strike price. The date is known as the expiration data, exercise date or maturity (Hull (2008)). Options on stocks were first traded on an organized exchange in 1973. Since then there has been a dramatic growth in the option markets. Options are now traded on many different exchanges throughout the world.

Not every option can be exercised at any time. European options can only be exercised at one given maturity date, whereas American options can be exercised at any time prior to their maturity date. "European" and "American" options are names for different types of the exercise right and not a geographical classification, i.e. American options can be traded in Europe and vice versa. The price of the option, also called premium, depends on the value of one entity of its underlying asset, which can be one or more stocks, an index (e.g. S&P500), a foreign currency, a future contract, etc. Due to this dependency on the underlying value, options belong to the group known as financial derivatives.

The option pricing model introduced by Black and Scholes was the first, and is still the most widely used, for the valuation of options. Black and Scholes observed a lognormal behavior of asset prices and derived a partial differential equation that describes the option's value.

The well-known Black-Scholes model for the option price is:

$$\frac{\partial}{\partial t}V(S,t)+\frac{1}{2}\sigma^2S^2\frac{\partial^2}{\partial S^2}V(S,t)+(r-\delta)S\frac{\partial}{\partial S}V(S,t)=rV(S,t), \tag{1}$$

where S is the asset price underlying the option, $S\geq 0$, V is the option price, r is the risk-free rate, t is the time since the option was issued, $0\leq t\leq T$, T is the time to maturity, δ is the dividend and σ is the volatility.

In other to solve this equation, it is necessary to set appropriate final and boundary conditions. These conditions depend on the option products. Different products have different conditions. For instance, the European Call and Put options have the payoff function (the final condition):

$$V(S,T)=\begin{cases} max(S-K,0), \\ max(K-S,0), \end{cases}$$

and the boundary conditions:

$$V(0,t) \approx \begin{cases} 0, \\ Ke^{-r(T-t)}, \end{cases}$$

$$V(S_{max},t) \approx \begin{cases} S, \\ 0, \end{cases}$$

where K is strike price and T is maturity date.

American options can be exercised before maturity date. The early exercise constraint means that the value of an American option at each time-step in the numerical computation is compared with the value of the payoff. When the value of the payoff is higher (it is optimal to exercise early), the value of the option is replaced by the value of payoff (Hull (2008); p.431).

In the case of the option on two underlying assets, the Black-Scholes equation is extended to two-dimensional equation:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r-\delta_1)S_1\frac{\partial V}{\partial S_1} + (r-\delta_2)S_2\frac{\partial V}{\partial S_2} = rV, \quad (2)$$

where ρ is the correlation of two assets.

Many types of two-asset options with various payoff functions are traded in financial markets. Some example payoffs include:

Options on a basket

$$V(S_1,S_2,T) = \begin{cases} max((S_1+S_2)-K,0), \\ max(K-(S_1+S_2),0), \end{cases}$$

Options on the minimum

$$V(S_1,S_2,T) = \begin{cases} max(min(S_1,S_2)-K,0), \\ max(K-min(S_1,S_2),0), \end{cases}$$

Options on the maximum

$$V(S_1,S_2,T) = \begin{cases} max(max(S_1,S_2)-K,0), \\ max(K-max(S_1,S_2),0), \end{cases}$$

These options are most commonly traded multi-asset options. Options on stocks are traded as mostly American style (Seydel (2009)). At present these options have no known analytical solution, therefore, these are usually priced with numerical methods. The study of efficient numerical approaches for option pricing with regard to the speed of computation and accuracy is still active field of research.

In the Section 3, the FEM has been applied to the Eq.1 and Eq.2 and numerical results by the FEM are shown in section 6.

3 Finite element applied to option pricing

With the finite element method, the method of weighted residuals is used to construct an integral formulation called a variational form. Several different weighted residual methods exist. We follow the popular method, Galerkin's method. This method is helpful in the computational effort as it produces systematic matrices. We also follow a common technique, called the method of lines, which uses finite elements in space and finite differences in time.

Eq.1 and Eq.2 are backward moving equations, i.e. these are solved from future to present. It might be convenient to replace the time variable t by the time to maturity $T-t$. Doing so, we get the linear parabolic equation forwards in time.

In 1D problem, Eq.1 is to be of the following form:

$$\frac{\partial}{\partial t}V(S,t) - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2}V(S,t) - (r-\delta)S\frac{\partial}{\partial S}V(S,t) = -rV(S,t). \tag{3}$$

To apply Galerkin's method, we first multiply Eq.3 by the weight function and minimize the residual, say R. The residual $R = L(v) - f$, where v is an approximate solution, L is the differential operator and f is a function of independent variables. The residual R is minimized to zero by weighting it with the so-called weight or test function $w(S)$.

This result in Eq.3 is

$$\int wRds = \int w(L(v)-f)ds = \int w(\frac{\partial v}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} - (r-\delta)S\frac{\partial v}{\partial S} + rv)ds = 0, \tag{4}$$

where integration is over the domain of interest.

With integration by parts to reduce the order of differentiation in the integrand, which allows the use of linear shape functions, Eq.4 is to be following variational form:

$$\int (\frac{\partial v}{\partial t} w + \frac{1}{2}\sigma^2 S^2 \frac{\partial v}{\partial S}\frac{\partial w}{\partial S} + \sigma^2 S \frac{\partial v}{\partial S} w - (r-\delta) S \frac{\partial v}{\partial S} w + rvw) ds = 0, \tag{5}$$

where the nonintegral terms become zero, with the proper choice of weight function.

For the Galerkin method, the approximate solution v and the weighting function w are given by

$$v(S,t) := \sum_{j=1}^{N} \alpha_j(t)\phi_j(S), \quad w(S) := \sum_{i=1}^{N} \phi_i(S), \tag{6}$$

where ϕ is the basis functions on a suitable mesh where N is the number of interior nodes.

We introduce Eq.6 into Eq.5 to obtain the system of N linear equations in the N unknown functions of time $\alpha_j(t)$ of the form:

$$\int \sum_{i=1}^{N} (\sum_{j=1}^{N} \dot{\alpha}_j \phi_j \phi_i + \frac{1}{2}\sigma^2 S^2 \int \sum_{j=1}^{N} \alpha_j \phi_j' \phi_i' + \sigma^2 S \sum_{j=1}^{N} \alpha_j \phi_j' \phi_i$$
$$- (r-\delta) S \sum_{j=1}^{N} \alpha_j \phi_j' \phi_i + r \sum_{j=1}^{N} \alpha_j \phi_j \phi_i) ds = 0. \tag{7}$$

The symbol “” denotes the temporal derivative and “” denotes the spatial derivative.

The Eq.7 can be rewritten as follows:

$$\sum_{j=1}^{N} [T_{ij}\dot{\alpha}_j + K_{ij}\alpha_j] = 0, i = 1,2,\ldots,N, \tag{8}$$

where $T_{ij} = \int \phi_j \phi_i ds, \quad K_{ij} = \int \frac{1}{2}\sigma^2 S^2 \phi_j' \phi_i' + r\phi_j\phi_i + (\sigma^2 S - (r-\delta)S)\phi_j'\phi_i) ds$,$i$, j= 1, 2, . . . , N.

The system of Eq.8 is a semi-discrete finite element approximation. To obtain a fully discrete approximation, for instance, we could use the explicit finite difference approximation,

$$\frac{d\alpha_j}{dt} \cong \frac{\alpha_j^{k+1} - \alpha_j^k}{\Delta t}, \alpha_j^k \equiv \alpha_j(k\Delta t).$$

To generalize, we use a weight average θ-scheme which is given by
$\alpha_j(t) \equiv (1-\theta)\alpha_j^k + \theta\alpha_j^{k+1}$, where $\theta \in [0, 1]$.
Then, the discrete scheme corresponding to Eq.8 is defined as follows:

$$\sum_{j=1}^{N}(\frac{1}{\Delta t}T_{ij} + \theta K_{ij})\alpha_j^{k+1} = \sum_{j=1}^{N}(\frac{1}{\Delta t}T_{ij} - (1-\theta)K_{ij})\alpha_j^k, \tag{9}$$

$i = 1, 2, \ldots, N, k = 1, 2, \ldots, M$,

where we partition the time domain $0 \le t \le T$ into M equal intervals of length $\Delta t = T/M$.

For θ=0, 1 and 1/2, we respectively obtain the explicit, the implicit, and the Crank-Nicolson schemes.

With the initial conditions and boundary conditions, depending on option products, the algebraic system Eq.9 is solved for each time t and the approximate solutions $V(S,t)$ are obtained.

When considering the two-dimensional Black-Scholes Eq.2, similar procedures are applied. Eq.6 is rewritten such as:

$$v(S_1, S_2, t) := \sum_{j=1}^{N} \alpha_j(t)\phi_j(S_1, S_2), \quad w(S_1, S_2) := \sum_{i=1}^{N} \phi_i(S_1, S_2).$$

And, T_{ij} and K_{ij}in Eq.8 are written such as:

$$T_{ij} = \iint \phi_j \phi_i dS_1 dS_2,$$

$$K_{ij} = \iint [\frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial \phi_j}{\partial S_1}\frac{\partial \phi_i}{\partial S_1} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial \phi_j}{\partial S_2}\frac{\partial \phi_i}{\partial S_2} + \frac{1}{2}\rho\sigma_1\sigma_2 S_1 S_2 (\frac{\partial \phi_j}{\partial S_1}\frac{\partial \phi_i}{\partial S_2} + \frac{\partial \phi_j}{\partial S_2}\frac{\partial \phi_i}{\partial S_1})$$
$$+ (\sigma_1^2 + \frac{1}{2}\rho\sigma_1\sigma_2 - (r-\delta_1)S_1)\frac{\partial \phi_j}{\partial S_1}\phi_i + (\sigma_2^2 + \frac{1}{2}\rho\sigma_1\sigma_2 - (r-\delta_2)S_2)\frac{\partial \phi_j}{\partial S_2}\phi_i$$
$$+ r\phi_j\phi_i] dS_1 dS_2.$$

More information about option pricing using the FEM can be found in Achdou and Pironneau (2005), Seydel (2009) and Topper (2005a, 2005b).

4 Development of an option pricing program

The actual finite element calculation starts to reduce the infinite number of degrees of freedom of a continuum problem to a finite number of unknowns, defined at element nodes. After the calculation of local element contributions, it ends with numerical solutions of the resulting system of algebraic equations.

In the procedures of finite element calculations, central part of computation is solving the system of equations derived from Eq.9. For this, we used our efficient solver, IPSAP (Kim, Lee and Kim (2002, 2005)). IPSAP (Internet Parallel Structural Analysis Program) is the general purpose finite element analysis software, developed by ASTL (Aerospace Structures Laboratory of Seoul National University). IPSAP solver has been developed focusing on large scale computations and considered distributed memory environments so it shows high performance with the standard PC, for general users, as well as with large-scale parallel computing system. It also shows outstanding performance in various platforms, from portable notebooks to a super computer. This has been verified through several international dissertations and presentations. IPSAP is published in Windows (x86, x64, serial and parallel), Linux (x86, x64, serial and parallel), and Mac OS (G4, G5, serial) versions and is available as a free download (http:// ipsap.snu.ac.kr).

To efficiently solve the Black-Scholes equation for option pricing, we have developed the option pricing program, embedding IPSAP solver. The superiority of performance of our program is shown in section 6.

The developed option pricing program has been implemented in C++ programming language, based on the object-oriented architecture. Since the option products have been developed with various boundary conditions, final conditions, number of assets, the option pricing program should be easily extensible. This is the reason the program has been designed according to object-oriented architecture. Additional benefits of this architecture are maintenance, flexibility and reusability (Kromer, Dufosse and Gueury (2005)).

Fig. 1 shows the class architecture of major classes with some methods and attributes for the finite element modeling of the option pricing, using UML (Unified Modeling Language). The UML is used for the description of relationships between classes.

In the Fig. 1, main purposes of class *FemModel* are reading data on the finite element model, storing this data, and providing it for the class *PDETimeParabolic*. Class *FemModel* has virtual functions, and these functions are implemented at class *Element*, *Node*, *OptionProperty*.

Class *Element* gets information from class *Node* and class *OptionProperty*. Class *BSPDE* associates with class *shapeFunction* and *GaussianQuadrature* to construct element matrices, using the inheritance of methods and properties of the class *Element*.

Class *FemModel* associates with class *PDETimeParabolic* through method *Fem-*

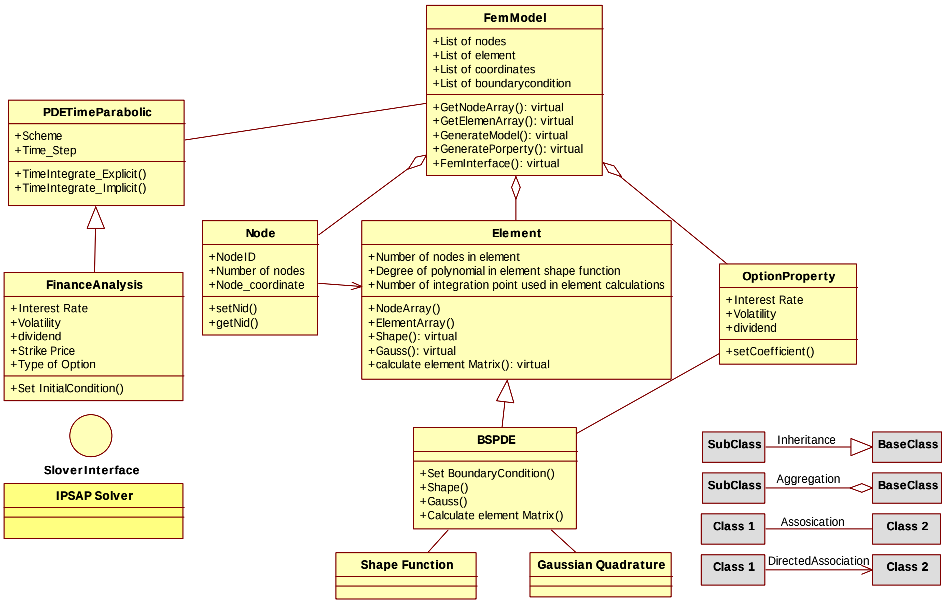

Figure 1: A brief class diagram of the FEM modeling for option pricing

Interface, to solve the system of equation at each time step. Class *FinanceAnalysis* inherits from class *PDETimeParabolic* and associates with class *IPSAPSolver* by method *SolverInterface*.

This class architecture is very suitable to handle various option products, by extending classes.

5 Self-made cluster computer system

Due to the cost advantage for building a parallel system, one current trend is building a self-made parallel computer in laboratories and academic departments. The preferred and common configuration could be a Linux cluster system (Paik, Moon and Kim (2006)). Our laboratory has developed the cluster computer system; called *Pegasus* (http://astl.snu.ac.kr/ENG/Research/pegasus01.asp?tp1=022). Pegasus consists of 260 nodes, 520 Intel Xeon 2.2/2.4/2.8/3.06 GHz processors. Each node in Pegasus runs dual Intel Xeon processors on the E7500 chipset motherboard with 2 (or 3, 4, 6) GB DDR RAM and 80/160 GB HDD. Gigabit network system was utilized for parallel computing. The Nortel Baystack 380-24T L2 switches were connected to 260 nodes and to Passport 8600 Routing switch at the core of Gigabit Ethernet network system. For cluster management and NFS

(Network File System) service, the Fast Ethernet network system was constructed. The NetPIPE(a Network Protocol Independent Performance Evaluator) test (Snell, Mikler, Gustafson and Helmer (1996)) was carried out to measure the network performance. The tuned-up network bandwidth of two nodes is about 920 Mbps and the latency time is about 21 μsec.

Our option pricing program was executed on this *Pegasus* cluster system. A specific configuration of used nodes is presented in Tab. 1.

Table 1: Configuration of used cluster system for option pricing

System	Red Hat Linux release 9 (Shrike)	108 nodes
CPU	Intel(R) XEON(TM) 2 CPU 2.20GHz	76 nodes
	Intel(R) XEON(TM) 2 CPU 2.40GHz	23 nodes
	Intel(R) XEON(TM) 2 CPU 2.80GHz	6 nodes
	Intel(R) XEON(TM) 2 CPU 3.06GHz	3 nodes
RAM	1 GB	1 node
	2 GB	1 node
	3 GB	95 nodes
	4 GB	3 node
	6 GB	8 node

6 Numerical results

To numerically study the performance, it is necessary to apply it to known option pricing problem found in the literature. The results are obtained and compared with known results. We simulate the valuations of option on a single node and then extend to the cluster computer system to handle a lot of valuations at once.

The numerical verification in this section shows the strength in computation time as well as accuracy.

6.1 Run on Single Node

We consider the European and American option and the corresponding Greeks. The European option doesn't need to solve numerically because it has an analytic solution. However, for showing the accuracy of our simulation, by comparison with the analytical solutions, we employed the European option.

For the numerical valuations in this section, a single node with Intel(R) XEON(TM) 2 CPU 2.20GHz, 3 GB DDR RAM and Red Hat Linux release 9 (Shrike) was used.

6.1.1 One underlying asset

Let's consider the numerical solution of one dimensional Black-Scholes partial differential equation. As we described in section 2, one dimensional equation means one underlying asset option products.

Tab. 2 shows results of the European call and put option pricing and Tab. 3 is for corresponding Greeks of European call option, compared with analytical solutions. Greek is the sensitivity measure of option from its parameters. We calculated three of the main Greeks: delta, gamma and theta. The Delta of option is defined as the rate of change of the option price with respect to the price of the underlying asset. Gamma is depending on the delta changes, that is, the rate of change of delta with respect to the asset price. The Theta of option is the rate of time decay for an option. All Greeks are hedge parameters. It is important to calculate, not only the option price itself, but also the Greeks in a quick and stable way since they are used when hedging the options. More information about Greeks can be found in Duffy (2006) and Hull (2008).

All results in this section were computed employing the linear shape functions

Table 2: The Valuation of the European options Maturity=1 year, Strike=$50, risk-free rate= 5%, Volatility=25%

S	Call Option Price		Put Option Price	
	Analytic	FEM	Analytic	FEM
46	3.91405	3.91367	5.47552	5.47532
50	6.16799	6.16713	3.72946	3.72928
54	8.90414	8.90185	2.46561	2.46544
RMSE		0.00143		0.00018
CPU(sec.)		0.268		0.253

Table 3: The Valuation of Greeks of the European call option Maturity=1 year, Strike=$50, risk-free rate= 5%, Volatility=25%

S	Delta		Gamma		Theta	
	Analytic	FEM	Analytic	FEM	Analytic	FEM
46	0.49660	0.49652	0.03469	0.03467	-0.00888	-0.00887
50	0.62741	0.62719	0.03027	0.03022	-0.00993	-0.00992
54	0.73658	0.73602	0.02419	0.02406	-0.01027	-0.01023
RMSE		0.00035		0.00008		0.00002
CPU(sec.)		0.365		0.370		0.457

(refer to Eq. 5). In the Tab. 2 and Tab. 3, the first column "S" means the underlying asset value (or stock price) and the row "RMSE" is Root Mean Square Errors. "CPU" means the elapsed time in seconds. The data under the column "Analytic" which means analytical solution are obtained from the Black-Scholes closed-form solution. The column "FEM" means the data calculated, using our option pricing program presented in section 4. This data are obtained using the explicit scheme with the time step 12,000 and considering the rage of stock price 10∼110 and number of nodes 400 on the uniform grid. Values are rounded off the numbers to six decimal places.

Tab. 4 is for showing the convergence in the case of underlying asset S=$50 of the Tab. 2.

Table 4: The convergence of European Put Option with underlying asset $50

Number of nodes	Time Steps	FEM	Error	Reduction Rate
25	2500	3.66645	0.06301	
50	10000	3.71395	0.01551	2.02238
100	40000	3.72560	0.00386	2.00653
200	160000	3.72851	0.00096	2.02260
400	640000	3.72923	0.00023	2.04629

In the Tab. 4, "Error" is the absolute value of the difference between analytical solution and FEM and "Reduction rate" is defined as:

$$\text{Reduction rate} \equiv log_2 \frac{|u_{\Delta S} - u_{exact}|}{\left|u_{\frac{\Delta S}{2}} - u_{exact}\right|},$$

where $u_{\Delta S}$ denotes the FEM solution with the spatial mesh size ΔS, and u_{exact} is the analytical solution.

We now consider the American option. Since the American option does not have analytical solutions, the binomial method is used as a benchmark to calculate RMSE, found in Wu and Kwok (1997) and Topper (2005b). The binomial method tends to be used only for research purpose because it is computationally so intensive. So, comparing elapsed time with this method is meaningless. Tab. 5 shows results of the American put option pricing and the corresponding Delta.

The data under the column "Binomial" are obtained using the binomial method with 1,000 time steps and "FEM" are obtained using the explicit scheme with the time step 14,000 and considering the range of stock price 20∼200, number of nodes 360 on the uniform grid. Option price and Delta values are rounded off the numbers to five decimal places and RMSEs are to six decimal places.

We can say that our option pricing program is very efficient and reliable.

Table 5: The Valuation of the American put option and Delta Maturity=1 year, Strike=$100, risk-free rate= 10%, Volatility=25%

S	Option Price		Delta	
	Binomial	FEM	Binomial	FEM
80	20.2687	20.2683	-0.8632	-0.8631
90	13.1228	13.1203	-0.5829	-0.5829
100	8.3348	8.3373	-0.3856	-0.3855
110	5.2091	5.2084	-0.2491	-0.2491
120	3.2059	3.2073	-0.1575	-0.1575
RMSE		0.00159		0.00008
CPU (sec.)		0.417		0.517

Next, we expend to multiple assets whose payoff is dependent on the values of several underlying assets. The two asset option pricing problems are visited.

6.1.2 Two underlying assets

Let's consider the option of two assets. Tab. 6 is the results for three types of options: European call options on the maximum of the two assets, European put options on the minimum of the two assets, and American put options on the minimum of two assets. The results are compared to the results of the analytical formula and to the results of the lattice method from Boyle (1988). The prices are rounded off the numbers to four decimal places and RMSEs are to six decimal places.

In this table, "S" means the strike price, "RMSE" is Root Mean Square Errors and "CPU" is the elapsed time in seconds. The data under the column "Analytical" (analytical solution) and "Boyle" were from Boyle (1988). Boyle's data were computed using lattice approach with 50 time steps. Our results ("FEM") were obtained using the range of stock price 10$\sim$90, number of nodes 120$\times$120 on the uniform girds, and time step 1,000 with explicit scheme.

To compare the elapsed time, we found Kargin (2004), which uses Boyle's data and shows the CPU times. However, difference computer language and specifications of computer made it difficult to compare fairly. Kargin (2004) used the interpolated lattice and irregular grid for Boyle's examples. His pricing was done on a standard IBM PC with Pentium microprocessor and Windows 2000 operating system. The entire code is written in Matlab. The pricing times for European and American Put on Minimum of two assets are shown as 8$\sim$10 minutes.

Until now, we saw that our option pricing program yields accurate solutions with speed-up computation. Although our program is very fast, if we need to calculate

Table 6: The Valuation of two-asset options Maturity=7 months, risk-free rate= 4.8790%, Correlation=0.5 first asset price=\$40, second asset price=\$40, Volatility for first asset=20%, Volatility for second asset=30%

European Call on the Maximum			
S	Analytical	Boyle	FEM
35	9.420	9.419	9.418
40	5.488	5.483	5.484
45	2.795	2.792	2.795
RMSE		0.00342	0.00258
CPU (Sec.)			3.26
European Put on the Minimum			
S	Analytical	Boyle	FEM
35	1.387	1.392	1.387
40	3.798	3.795	3.794
45	7.500	7.499	7.497
RMSE		0.00342	0.00289
CPU (Sec.)			3.26
American Put on the Minimum			
S		Boyle	FEM
35		1.423	1.418
40		3.892	3.891
45		7.689	7.691
CPU (Sec.)			4.226

50 option prices, for example American put on the minimum of two assets described Tab. 6, it takes more than 200 sec. Even if we can endure the computation time, it is too much of a hassle to do job one by one. In the next section, we extend to cluster system and handle a lot of valuations at once.

6.2 Run on Cluster System

We now consider the implementation on the cluster system. When the cluster system is applied to the finance models, the advantages in two ways could be considered. First, this system can give the possibility to treat huge number of option products with low cost of installation and maintenance. Many finance companies and laboratories are using supercomputers or GPU systems to analyze worldwide market option products. Instead of those expensive systems, the cluster system would be an attractive alternative to handle easily large amount of data. Second, the analyses of option prices and Greeks on various conditions are available, since option prices with respect to different volatilities, interest rates, correlations and maturity dates could be calculated at once. These analyses allow faster reaction to market conditions, enabling to take into account larger data.

The possibilities of those two advantages were reflected in the below implementation. Our option pricing program executed on the cluster system, using data of the American put on the minimum of two assets in Tab. 6.

The results of option pricing on the cluster system were visualized as the 2D plot in Fig. 2 and 3D plot in Fig. 3.

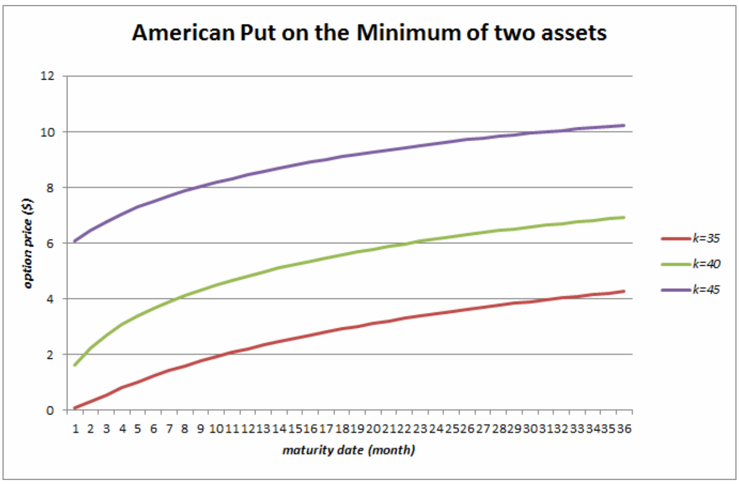

Figure 2: Option pricing on the cluster system with respect to the change of maturities

Fig. 2 shows the numerical solutions of the option pricing with respect to the change of the maturities from 1 month to 36 months for each strike price (K=35,40,45), using the 108 nodes in Tab. 1.

The total elapsed time of the computation for 108 option prices is 15.024 seconds (One option pricing on the single node is 4.226 sec.) like below:

```
[sjkim@n081 ksiam]$ time sh runPricing.sh  > data.txt

real    0m15.024s
user    0m0.130s
sys     0m0.270s
```

The American put options become more valuable as the time to expiration increases, as we know. But, we can see that the change in the price of the option

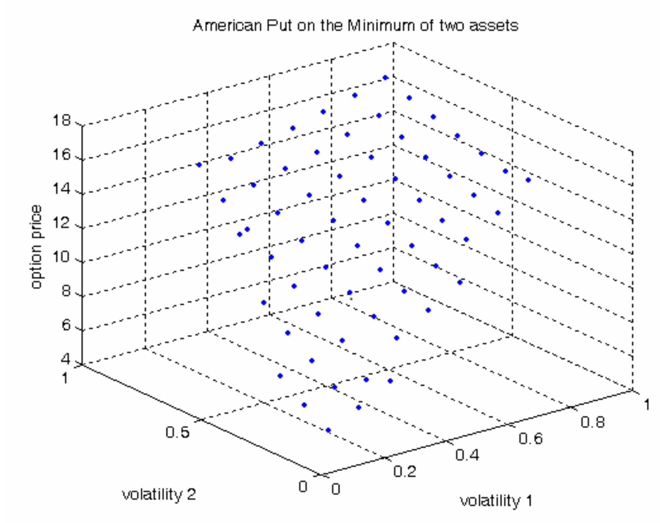

Figure 3: Option pricing on the cluster system with respect to the change of volatilities

decreases, in Fig. 2, which could be very helpful to choose the optimal investment time.

Fig. 3 shows option prices with respect to the change of volatilities from 10% to 90% (strike price is $45 and maturity is 7 months).

Volatility is the key to understanding why option prices act the way they do. It is the most important concept in the options analysis. Our results in Fig. 3 could help get a handle on the relationship of volatilities to most options strategies.

We can infer that using the cluster computer system is very efficient and economical for the analyses of option pricing models.

7 Conclusions

We have proposed and tested the option pricing program with the finite element method and simulated it on the distributed computing system that is a self-made cluster computer system. The novelty of our approach is not only solving option pricing models accurately and efficiently, but also using the relatively cheap and easy to implement clustering.

In the first part of this paper, we introduced the option pricing model applied to the finite element technique and presented the developed option pricing program and self-made Linux cluster system. The second part was devoted to the numerical simulations to demonstrate the superiority of our approach. Our option pricing program gives option prices and the corresponding Greeks. The numerical solutions of this program are accurate and the computation times are very positive, without additional numerical techniques or algorithms. This program executed on the cluster computer system, which is the very economical way, handle a lot of valuations of option at once.

The option pricing on the distributed computing could afford the means of investigating how the pricing models change under various conditions. The model parameter estimation based on this investigation would drive improving the accuracy of finance model and this is a very important issue in the field of financial research.

In conclusion, our research is a natural fit for running the numerous simulations to efficiently analyze the valuations in stock portfolios, financial products, or other investment vehicles. We could expect it brings new opportunities for pricing the entire portfolios, analyzing the large scale models or performing the market comparisons. Also, it would be very useful to analyze the exotic option pricing models including complex strategic decisions.

Acknowledgement: This work was supported by the National Space Lab program of National Research Foundation of Korea. (Contract No. 2009-0092052).

References

Achdou, Y.; Pironneau, O. (2005): *Computational Methods for Option Pricing*. SIAM, Philadelphia.

Black, F.; Scholes, M. (1973): The Pricing of Options and Corporate Liabilities. *J. Polit. Eco.,* vol. 81, pp. 637–654.

Boyle, P. P. (1988): A lattice framework for option pricing with two state variables. *J. Financ. Quant. Anal.*, vol. 23, no. 1, pp. 1–12.

Broadie, M.; Detemple, J. (1996): American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods. *Rev. Financ. Stud.,* vol. 9, pp. 1211–1250.

Choi, S.; Marcozzi, M. D. (2004): On the application of MQ-RBF to the valuation of derivative securities. *CMES: Computer Modelling & Engineering Sciences,* Vol. 5, No. 3, pp. 201-212.

Cooney, M. (2000): Report on the Accuracy and Efficiency of the Fitted Methods for Solving the Black-Scholes Equation for European and American Options. *URL: http://www.datasimfinancial.com/UserFiles/articles/BlackScholesReport2.ps.*

Duffy, D. J (2006): *Finite difference methods in financial engineering: a Partial*

Differential Equation Approach. Wiley Finance.

Foufas, G.; Larson, M. G. (2008): Valuing Asian options using the finite element method and duality techniques. *J. Comput. Appl. Math.*, vol. 222, pp. 144–158.

Hull, J. (2008): *Options, Futures, and other Derivatives*. Prentice Hall, 7^{th} Pearson International edition.

Jackson, N.; Süli, E. (1997): Adaptive finite element solution of 1D option pricing problems. *Oxford University Computing Laboratory*, Report 97/05.

Kargin, V. (2004): Lattice option pricing by multidimensional interpolation. *Math. Finance*, vol. 15, no. 4, pp. 635–647.

Khaliq, A. Q. M.; Voss, D. A.; Kazmi, K. (2006): A Linearly Implicit Predictor-Corrector Scheme for Pricing American Options using a Penalty Method Approach. *J. J. Bank. Finance.*, vol. 30, pp. 489-502.

Khaliq, A. Q. M.; Voss, D. A.; Kazmi, K. (2008): Adaptive θ-methods for pricing American options. *J. Comput. Appl. Math.*, vol. 222, pp. 210–227.

Kim, S.J.; Lee, C.S.; Kim, J.H. (2002): Large-Scale Structural Analysis by parallel multifrontal solver through Internet Based PCs. *AIAA Journal*, vol. 40, No.2, pp. 359-367.

Kim, S.J.; Lee, C.S.; Kim, J.H. (2005): High-Performance Domainwise Parallel Direct Solver for Large-Scale Structural Analysis. *AIAA Journal*, vol. 43, pp. 662-670.

Kromer, V.; Dufosse, F.; Gueury, M. (2005): On the implementation of object-oriented philosophy for the design of a finite element code dedicated to multibody systems. *Finite Elem. Anal. Des.*, vol. 41, pp. 493-520.

Nielsen, B. F.; Skavhaug, O; Tveito, A. (2002): Penalty and front-fixing methods for the numerical solution of American option problems. *J. Comput. Finance, Vol. 5, pp. 69-97.*

Nielsen, B. F.; Skavhaug, O; Tveito, A. (2008): Penaltymethods for the numerical solution of American multi-asset option problems. *J. Comput. Appl. Math., Vol. 222, pp. 3-16.*

O'sullivan, S.; O'sullivan, C. (2011): On the acceleration of explicit finite difference methods for option pricing. *Quant. Finance*, vol. 11, no. 8, pp. 1177-1191.

Paik, S. H.; Moon, J. J.; Kim, S. J. (2006): Parallel performance of large scale impact simulations on Linux cluster super computer. *Comput. Struct*, vol. 84, pp. 732-741.

Rogers, L. C. G. (eds.)**; Talay, D.**(eds.) (1997): *Numerical Methods in Finance*. Cambridge University Press.

Sapariuca, I.; Marcozzib, M. D.; Flahertyc, J. E. (2004): A numerical analysis of variational valuation techniques for derivative securities. *Appl. Math. Comput.*, vol. 159, pp. 171–198.

Seydel, R. (2009): *Tools for Computational Finance*. Springer, 4^{th} edition.

Snell, Q.; Mikler, A.; Gustafson, J; Helmer, G (1996): NetPIPE: a network protocol independent performance evaluator. *In: Proceedings of the IASTED international conference*

Suh, S. (2009): Pseudospectral methods for pricing options. *Quant. Finance*, vol. 9, no. 6, pp. 705–715.

Tomas III, M. J.; Yalamanchili, K. K. (2001): An Application of Finite Elements to Option Pricing. *J. Futures Market.*, vol. 21, no. 1, pp. 19–42.

Topper, J. (2005a): *Financial Engineering with Finite Elements*. Wiley Finance.

Topper, J. (2005b): *Option Pricing with Finite Elements*. Wilmott magazine, January.

Wallner, C.; Wystup, U. (2004): Efficient Computation of Option Price Sensitivities for Options of American Style. *HfB - Business School of Finance & Management*, No. 1.

Wu, L.; Kwok, Y. K. (1997): A front-fixing finite difference method for the valuation of American options. *The journal of Financial Engineering*, vol. 6, no. 2, pp. 83–97.

Young, D. L.; Sun, C. P.; Shen, L. H. (2009): Pricing Options with Stochastic Volatilities by the Local Differential Quadrature Method. *CMES: Computer Modelling & Engineering Sciences,* Vol. 46, No. 2, pp. 129-150.

Zhu, W.; Kopriva, D. A. (2010): A Spectral Element Approximation to Price European Options with One Asset and Stochastic Volatility. *J. Sci. Comput.*, vol. 42, pp. 426–446.

Zvan, R.; Forsyth, P. A.; Vetzal, K. R. (1998): Penalty methods for American options with stochastic volatility. *J. Comput. Appl. Math.*, vol. 91, pp. 199–218.