J. Korean Soc. Ind. Appl. Math. Vol.28, No.3, 88–95, 2024 https://doi.org/10.12941/jksiam.2024.28.088

AN EFFICIENT AND ACCURATE ADAPTIVE TIME-STEPPING METHOD FOR THE BLACK–SCHOLES EQUATIONS

HYEONGSEOK HWANG[1], SOOBIN KWAK[2], YUNJAE NAM[3], SEOKJUN HAM[2], ZHENGANG LI[3], HYUNDONG KIM[4], AND JUNSEOK KIM[2†]

[1]DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 02841, REPUBLIC OF KOREA

[2]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 02841, REPUBLIC OF KOREA

[3]PROGRAM IN ACTUARIAL SCIENCE AND FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 02841, REPUBLIC OF KOREA

[4]DEPARTMENT OF MATHEMATICS AND PHYSICS, GANGNEUNG-WONJU NATIONAL UNIVERSITY, GANGNEUNG 25457, REPUBLIC OF KOREA
Email address: [†]cfdkim@korea.ac.kr

ABSTRACT. In this article, we propose an efficient and accurate adaptive time-stepping numerical method for the Black–Scholes (BS) equations. The numerical scheme used is the finite difference method (FDM). The proposed adaptive time-stepping computational scheme is based on the maximum norm of the discrete Laplacian values of option values on a discrete domain. Most numerical solvers for the BS equations require a small time step when there are large variations in the solutions. To resolve this problem, we propose an adaptive time-stepping algorithm that uses a small time step size when the maximum norm of the discrete Laplacian values on a discrete domain is large; otherwise, a larger time step size is used to speed up the computation. To demonstrate the high performance of the proposed adaptive time-stepping methodology, we conduct several computational experiments. The numerical tests confirm that the proposed adaptive time-stepping method improves both the efficiency and accuracy of computations for the BS equations.

1. INTRODUCTION

We propose an efficient and accurate adaptive time-stepping numerical algorithm for the Black–Scholes (BS) equations:

$$\frac{\partial V(S,t)}{\partial t} = -\frac{1}{2}(\sigma S)^2\frac{\partial^2 V(S,t)}{\partial S^2} - rS\frac{\partial V(S,t)}{\partial S} + rV(S,t), \text{ for } S \geq 0,\ t > 0, \tag{1.1}$$

Received July 18 2024; Revised September 20 2024; Accepted in revised form September 24 2024; Published online September 25 2024.

Key words and phrases. Adaptive time-stepping, Discrete Laplacian, Black–Scholes equation.

[†] Corresponding author.

where S is the value of the underlying index, t is time, σ is a constant volatility of the underlying index, $r > 0$ is a constant risk-free interest rate, and $V(S,t)$ is the option value at $V(S,t)$. The terminal condition is the payoff function $\Lambda(S)$ at the expiration time T:

$$V(S,T) = \Lambda(S). \tag{1.2}$$

Finding the analytical solutions of Eqs. (1.1) and (1.2) for complex options such as equity-linked securities (ELS) is difficult. Therefore, using a numerical approximation is essential. To estimate the option value, one can compute a solution to the BS Eqs. (1.1) and (1.2) using the finite difference method [1, 2, 3], finite element method [4, 5, 6], finite volume method [7, 8], or fast Fourier transform [9, 10, 11]. Solutions to the BS equations often show both fast and slow time changes. For instance, the solution may change rapidly if the price of the underlying asset changes rapidly, in which case a small time step size may be needed to obtain an accurate solution. Conversely, if the change is slow, a larger time step size can be used. An adaptive time-stepping scheme automatically detects these changes and dynamically adjusts the time step size to increase computational efficiency [12]. A space-time adaptive finite difference method was developed in [13]. In [14], to price American options under constant or stochastic volatility assumptions, the authors considered adaptive time-stepping based on the second order backward difference formula [16] with variable time step sizes. To accurately and efficiently compute European cash-or-nothing call options and the Greeks near maturity, an adaptive time-stepping numerical algorithm was proposed in [17].

In this study, we use the finite difference method (FDM). The proposed adaptive time-stepping computational scheme relies on the maximum norm of discrete Laplacian values of option prices within a discrete domain. Many numerical solvers for the BS equations necessitate a small time step in the presence of significant solution variations. To resolve this issue, we introduce an adaptive time-stepping method that uses a small time step when the maximum norm of discrete Laplacian values is large within the discrete domain; otherwise, a larger time step is used to accelerate the computation. The proposed adaptive time-stepping scheme, based on discrete Laplacian values, is robust, efficient, simple, and accurate.

The main purpose of this paper is to develop an adaptive time-stepping computational scheme using the maximum norm of discrete Laplacian values of option prices.

This paper is organized as follows. In Section 3, the proposed efficient and accurate adaptive time-stepping numerical method for the BS equations is described. To demonstrate the high performance of the proposed adaptive time-stepping methodology, we conduct several computational experiments in Section 4. The numerical tests confirm that the proposed adaptive time-stepping method significantly improves both the efficiency and accuracy of the computations for the BS equations. Finally, Section 5 presents the conclusions.

2. Previous adaptive time-stepping methods

Now, we review several previous adaptive time-stepping schemes. In a previous study [17], with the fully implicit scheme employed for the time derivative, the numerical solution of Eq. (3.2) exhibits first-order accuracy with respect to time. The adaptive time-stepping strategy [18] was considered, where the time step is chosen based on criteria related to truncation error. To avoid excessively large or small time steps, the initial time step size $\Delta\tau_0$ is set with maximum and minimum time step sizes, $\Delta\tau_{\max}$ and $\Delta\tau_{\min}$. For the time step scaled error defined as

$$E_{\text{tr}} = \left\| u^{n+2} - v^{n+2} \right\|_2,$$

if the error is smaller than the given tolerance, the numerical solution at $(n+2)$-th step is set to u^{n+2}. Otherwise, the numerical solution is solved using time step $\Delta\tau_0/2$. Here, u^{n+2} and v^{n+2} are the numerical solutions obtained by using different time steps, $\Delta\tau_0$ and $2\Delta\tau_0$, respectively, and the scaled error is checked again. In this method, the next time step size is automatically determined by the given tolerance tol and the error E_{tr} as $\Delta\tau_{\text{new}} = \Delta\tau_0 \times tol/E_{\text{tr}}$.

In [19], an adaptive time-stepping method was proposed with the constraint that the maximum allowable time step $\Delta t_{\max}$ satisfies the boundedness of the numerical solutions. First, given $(n+1)$-th numerical solution v^{n+1} and the adaptive time step τ_n, compute v^n. For given tolerance δ, if $\|v^n - v^{n+1}\|$, then accept v^n and take a new time step as

$$\tau_{n-1} = \min\left(\tau_n \frac{\delta}{\|v^n - v^{n+1}\|}, \Delta t_{\max}\right).$$

Otherwise replace τ_n by $\tau_n/2$.

Persson and Sydow [14] developed an adaptive space-time finite difference method for pricing American options. The adaptive space is based on the Richardson extrapolation. The local discretization error is estimated on a coarse grid using two different step sizes. The adaptive time-stepping method is based on the work in [15]. The local time discretization with the two-step backward differentiation formula (BDF2) method is estimated at each time step by comparing the solution obtained with BDF2 to that obtained with an explicit scheme.

3. Computational method

Let us rewrite Eq. (1.1) as follows:

$$\frac{\partial V(x,\tau)}{\partial \tau} = \frac{1}{2}(\sigma x)^2 \frac{\partial^2 V(x,\tau)}{\partial x^2} + rx\frac{\partial V(x,\tau)}{\partial x} - rV(x,\tau), \text{ for } x \geq 0,\ \tau > 0, \tag{3.1}$$

where $x = S$ and $\tau = T - t$. Then, the initial condition is given as $V(x,0) = \Lambda(x)$. We discretize a numerical domain $\Omega = [0, L]$ as $\Omega_h = \{x_i | x_i = ih, \text{ for } i = 0, \ldots, N_x\}$, where $h = L/N_x$, see Fig. 1 for a schematic illustration.

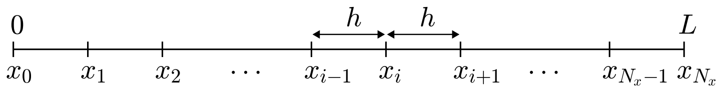

FIGURE 1. Discrete computational domain

Let $V_i^n = V(x_i, \tau_n)$, for $0 \leq i \leq N_x$ and $0 \leq n \leq N_t$, where $x_i = ih$ and $\tau_n = n\Delta\tau$. We discretize Eq. (3.1) as follows:

$$\frac{V_i^{n+1} - V_i^n}{\Delta\tau} = \frac{\sigma^2 x_i^2}{2} \frac{V_{i-1}^{n+1} - 2V_i^{n+1} + V_{i+1}^{n+1}}{h^2} + rx_i \frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2h} - rV_i^{n+1}, \tag{3.2}$$

which can be rewritten as

$$a_i V_{i-1}^{n+1} + b_i V_i^{n+1} + c_i V_{i+1}^{n+1} = f_i,$$

where $a_i = \dfrac{rx_i}{2h} - \dfrac{\sigma^2 x_i^2}{2h^2}$, $b_i = \dfrac{1}{\Delta\tau} + \dfrac{\sigma^2 x_i^2}{h^2} + r$, $c_i = -\dfrac{rx_i}{2h} - \dfrac{\sigma^2 x_i^2}{2h^2}$, and $f_i = \dfrac{V_i^n}{\Delta\tau}$. For the boundary conditions, we use $V_0^{n+1} = 0$ and $V_{N_x+1}^{n+1} = 2V_{N_x}^{n+1} - V_{N_x-1}^{n+1}$. Therefore, we have a tridiagonal system as follows:

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a_{N_x-1} & b_{N_x-1} & c_{N_x-1} \\ 0 & \dots & 0 & a_{N_x} - c_{N_x} & b_{N_x} + 2c_{N_x} \end{pmatrix} \begin{pmatrix} V_1^{n+1} \\ V_2^{n+1} \\ \vdots \\ V_{N_x-1}^{n+1} \\ V_{N_x}^{n+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N_x-1} \\ f_{N_x} \end{pmatrix}.$$

Let $V^n = (V_0^n,\ V_1^n, \dots, V_{N_x}^n)$. Furthermore, we define the discrete Laplacian as follows:

$$\left(\frac{\partial^2 V}{\partial x^2}\right)_i^n = \frac{V_{i-1}^n - 2V_i^n + V_{i+1}^n}{h^2}, \text{ for } i = 1, \dots, Nx.$$

Subsequently, we define the maximum norm of the discrete Laplacian $\|\partial^2 V^n/\partial x^2\|_\infty$ as the following.

$$\left\|\frac{\partial^2 V^n}{\partial x^2}\right\|_\infty = \max_{1\le i\le N_x} \left|\left(\frac{\partial^2 V}{\partial x^2}\right)_i^n\right|$$

Then, the proposed adaptive time-stepping method is described in the following algorithm:

Algorithm 1 An adaptive time-stepping method based on a discrete Laplace operator

Require: Set the initial condition V^0, the expiry time T, the maximum and minimum time steps $\Delta\tau_{\max}$ and $\Delta\tau_{\min}$, scaling factor s, time $\tau = 0$, iteration number $n = 0$.

```
while τ < T do
    Set Δτ = max[Δτ_min, τ_max/(1 + s‖∂²V^n/∂x²‖_∞)].
    if τ + Δτ > T then
        Set Δτ = T − τ.
    end if
    Solve Eq. (3.2) with the time step Δτ and V^n to obtain V^{n+1}.
    Set τ = τ + Δτ and n = n + 1.
end while
```

4. NUMERICAL TESTS

To demonstrate the high performance of the proposed adaptive time-stepping methodology, several computational experiments are conducted. The primary purpose of these experiments is to validate the efficiency and accuracy of the methodology under various payoff functions. The computational results consistently show improved accuracy and computational efficiency compared to traditional adaptive time-stepping methods. This improvement highlights the potential of this adaptive time-stepping approach for solving complex problems more effectively.

4.1. **Convergence tests.** We compute a European call option, where the payoff function is defined as follows:

$$V(x,0) = \max(x-K,0).$$

We have analytic solutions for these options. For the European call option, it is well known that the analytic solution of the BS equation is given by

$$V(x,\tau) = xN(d_1) - Ke^{-r\tau}N(d_2),\ \forall x \in [0,L],\ \forall \tau \in [0,T],$$
$$d_1 = \frac{\ln(x/K) + \left(r+\frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}},\ d_2 = d_1 - \sigma\sqrt{\tau},$$

where $N(d) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d} \exp\left(-\frac{x^2}{2}\right)dx$ represents the cumulative distribution function of the standard normal distribution. The convergence error is defined as:

$$\|err\|_2 = \sqrt{\frac{1}{N_x}\sum_{i=1}^{N_x}\left(V_i^n - V(x_i, n\Delta t)\right)^2}.$$

We used the following parameters: $S_0 = 100$, $K = 100$, $L = 3S_0$, $\sigma = 0.3$, $r = 0.02$, $\Delta t = 2^{-20}$, $T = 1$. Table 1 lists the l_2-norm of the error, $\|err\|_2$, and the convergence rates in the spatial dimension. The computational results demonstrate that the numerical scheme is second-order accurate with respect to the underlying asset.

TABLE 1. Errors and spatial convergence rates

N_x	15		30		60		120
$\|err\|_2$	0.24486		0.05936		0.01474		0.00369
rate		2.04		2.01		2.00	

The errors and convergence rates in the temporal dimension are listed in Table 2. Here, $N_x = 300$ is used. The numerical results validate that the algorithm is first-order accurate with respect to the temporal space due to the use of the fully implicit Euler scheme.

Moreover, according to the study by Linde et al., adaptive time-stepping using an explicit predictor and the second-order accurate backward differentiation formula (BDF2) corrector improves convergence by adjusting step sizes based on local error estimation [20].

TABLE 2. Temporal convergence error and convergence rate

Δt	0.25		0.125		0.0625		0.0625
$\|err\|_2$	0.13427		0.06993		0.03588		0.01835
rate		0.94		0.96		0.97	

4.2. **Temporal evolution of** $\|\partial^2 V^n/\partial x^2\|_\infty$ **and** $1/(1+s\|\partial^2 V^n/\partial x^2\|_\infty)$**.** We consider a European call option with the following parameters: $T = 1$, $\Delta\tau = T/360$, $L = 300$, $N_x = 100$, $r = 0.03$, $\sigma = 0.3$, and $K = 100$. Figures 2(a) and (b) show the temporal evolution of $\|\partial^2 V^n/\partial x^2\|_\infty$ and $1/(1+s\|\partial^2 V^n/\partial x^2\|_\infty)$, respectively. We can observe that $\|\partial^2 V^n/\partial x^2\|_\infty$ is decreasing with respect to time; and $1/(1+s\|\partial^2 V^n/\partial x^2\|_\infty)$ are increasing with respect to time and the scaling factor s.

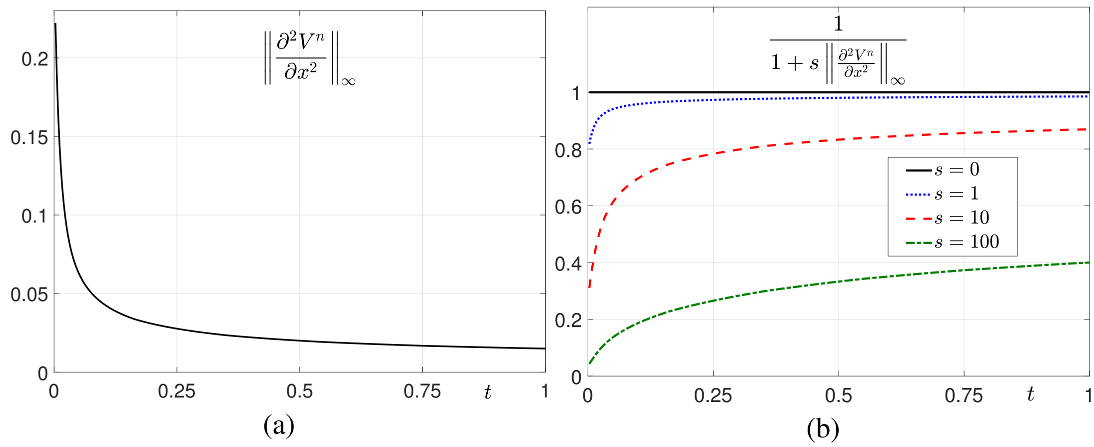

FIGURE 2. (a) and (b) represent the temporal evolution of $\|\partial^2 V^n/\partial x^2\|_\infty$ and $1/(1+s\|\partial^2 V^n/\partial x^2\|_\infty)$, respectively.

4.3. **Numerical prices for different scaling factors and** $\Delta\tau_{\max}$**.** We evaluate the European call option with the parameters $T = 1$, $L = 300$, $N_x = 1600$, $r = 0.03$, $\sigma = 0.3$, and $K = 100$. Figure 3 illustrates numerical prices at $S_0 = 100$ for different scaling factors and $\Delta\tau_{\max}$. For $\Delta\tau_{\max}$, Figs. 3(a) and (b) are set $\Delta\tau_{\min} = \Delta\tau_{\max}/2$ and $\Delta\tau_{\min} = \Delta\tau_{\max}/20$, respectively. From these results, we can confirm that the numerical solutions are more accurate when the scaling factor s is larger.

5. CONCLUSIONS

In this study, we introduced an efficient and accurate adaptive time-stepping numerical method for solving the BS equations. By using the FDM with an adaptive time-stepping strategy based on the maximum norm of the discrete Laplacian values within a discrete domain, we efficiently resolved the limitations of traditional numerical solvers, which require small time

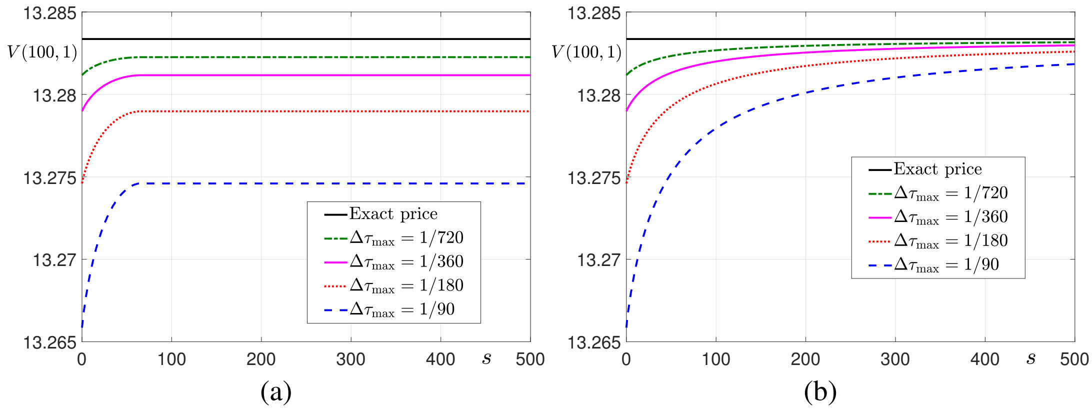

FIGURE 3. Numerical prices for different values of s and $\Delta\tau_{\max}$.

steps during periods of significant solution variations. Our adaptive approach dynamically adjusts the time step size, using smaller steps when the maximum norm of the discrete Laplacian values is large and larger steps otherwise, and thereby optimizes computational efficiency.

The results from our computational experiments validate the effectiveness of the proposed method. The adaptive time-stepping approach not only provides the accuracy of the solutions but also significantly improves computational efficiency. These findings suggest that the proposed methodology is a robust and reliable tool for numerically solving the BS equations and provides substantial improvements over conventional fixed time-stepping methods. Future work may explore the application of this adaptive strategy to other types of financial models and extend its implementation to more complex, multi-dimensional problems.

ACKNOWLEDGMENTS

The corresponding author (J.S. Kim) was supported by the Brain Korea 21 FOUR through the National Research Foundation of Korea funded by the Ministry of Education of Korea.

REFERENCES

[1] J. Wang, S. Wen, M. Yang, W. Shao, *Practical finite difference method for solving multi-dimensional Black-Scholes model in fractal market*, Chaos, Solitons & Fractals, **157** (2022), 111895.

[2] M.M. Khalsaraei, A. Shokri, H. Ramos, Z. Mohammadnia, P. Khakzad, *A positivity-preserving improved nonstandard finite difference method to solve the Black–Scholes equation*, Mathematics, **10(11)** (2022), 1846.

[3] N. Abdi, H. Aminikhah, A.H.R. Sheikhani, *High-order compact finite difference schemes for the time-fractional Black–Scholes model governing European options*, Chaos, Solitons & Fractals, **162** (2022), 112423.

[4] Y. Chen, L. Wei, S. Cao, F. Liu, Y. Yang, Y. Cheng, *Numerical solving for generalized Black–Scholes–Merton model with neural finite element method*, Digital Signal Processing, **131** (2022), 103757.

[5] E.R. Wijayanti, N. Susyanto, *European call options pricing numerically using finite element method*, IAENG International Journal of Applied Mathematics, **52(4)** (2022), 1–6.

[6] Anderson, T. (2023). *Numerical solution of the Black–Scholes equation using finite element methods*, Ph.D. thesis, Colorado State University.

[7] N. Thakoor, D.K. Behera, *A numerical algorithm based on a fitted finite volume method for option pricing with its implementation in mathematica*, ICT Systems and Sustainability, Lecture Notes in Networks and Systems, Springer, Proceedings of ICT4SD, Singapore 2022.

[8] M.A.H. Sajib, A.S. Hossain, S. Sudha, *A comparative analysis of Crank–Nicolson Scheme between finite volume method and finite difference method for pricing* European Option. GANIT: Journal of Bangladesh Mathematical Society, **42(2)** (2022), 1–13.

[9] Ewen, T. *Pricing of European calls with the quantum Fourier transform*, arXiv preprint (2024) arXiv:2404.14115.

[10] F. Baschetti, G. Bormetti, S. Romagnoli, P. Rossi, *The SINC way: A fast and accurate approach to Fourier pricing*, Quantitative Finance, **22(3)** (2022), 427–446.

[11] M. Ma, J. Yang, R. Liu, *A novel structure automatic-determined Fourier extreme learning machine for generalized Black—Scholes partial differential equation*, Knowledge-Based Systems, **238** (2022), 107904.

[12] R. Zvan, K.R. Vetzal, P.A. Forsyth, *PDE methods for pricing barrier options*, Journal of Economic Dynamics and Control, **24(11-12)** (2000), 1563-1590.

[13] J. Persson, L. von Persson, *Pricing European multi-asset options using a space-time adaptive FD-method*, Computing and Visualization in Science, **10** (2007), 173–183.

[14] J. Persson, L. von Sydow, *Pricing American options using a space-time adaptive finite difference method*, Mathematics and Computers in Simulation, **80(9)** (2010), 1922–1935.

[15] P. Lötstedt, J. Persson, L. von Sydow, J. Tysk, *Space–time adaptive finite difference method for European multi-asset options* Computers & Mathematics with Applications, **53(8)** (2007), 1159–1180.

[16] E. Hairer, S.P. Norsett, G. Wanner. *Solving ordinary differential equations*, Springer Series in Computational Mathematics 8, Springer-Verlag, Berlin, 1993.

[17] D. Jeong, M. Yoo, J. Kim, *Accurate and efficient computations of the greeks for options Near expiry using the Black–Scholes equations*, Discrete Dynamics in Nature and Society, **2016(1)** (2016), 1586786.

[18] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. *Flannery numerical recipes in C* (Vol. 2), Cambridge University Press, Cambridge, UK, 1996.

[19] A.Q. Khaliq, D.A. Voss, K. Kazmi, *Adaptive θ-methods for pricing American options*, Journal of Computational and Applied Mathematics, **222(1)** (2008), 210–227.

[20] G. Linde, J. Persson, L. Von Sydow, *A highly accurate adaptive finite difference solver for the Black–Scholes equation*, International Journal of Computer Mathematics, **86(12)** (2009), 2104–2121.