International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I11P219

Volume 71 Issue 11, 181-189, November 2023
© 2023 Seventh Sense Research Group®

*Original Article*

# Development of Red-Black Gauss-Seidel Algorithm for Efficiently Pricing Fixed Strike Asian Options based on Arithmetic Average

Wei Sin Koh[1], Saiful Hafizah Jaaman[2], Rokiah Rozita Ahmad[3], Jumat Sulaiman[4]

[1,2,3]*Department of Mathematical Sciences, University Kebangsaan Malaysia, Selangor, Malaysia.*
[1]*Faculty of Business and Communications, INTI International University, Negeri Sembilan, Malaysia.*
[4]*Faculty of Science and Natural Resources, University Malaysia Sabah, Sabah, Malaysia.*

[2]*Corresponding Author : shj@ukm.edu.my*

Received: 18 July 2023 Revised: 11 September 2023 Accepted: 12 October 2023 Published: 04 November 2023

***Abstract** - In this paper, the Red-Black Gauss-Seidel (RBGS) algorithm is developed to solve the arithmetic Asian option pricing. Developing such an algorithm is crucial for optimizing computational resources and reducing the processing time of the financial instrument. The pricing of arithmetic Asian options is formulated by approximating the Black-Scholes Partial Differential Equation (PDE) through the Crank-Nicolson finite difference method. Subsequently, the RBSG iterative algorithm is employed to solve the system of linear equations derived from the Crank-Nicolson approximation. Extensive computational experiments are conducted to measure the accuracy and efficiency of the RBGS algorithm to the conventional Gauss-Seidel (GS) iterative method. The evaluation criteria include the iteration count, computational time, and root mean squared error (RMSE). The results indicate that the RBSG iterative algorithm significantly reduces the number of iterations and computational time compared to the GS iterative method. Moreover, both iterations yield accurate numerical solutions that align closely. These findings demonstrate the effectiveness of the RBSG algorithm in efficiently pricing arithmetic Asian options while maintaining a high level of accuracy.*

***Keywords** - Asian option, Black-Scholes PDE, Crank-Nicolson finite difference method, Red-Black Gauss-Seidel algorithm, Resource efficiency.*

## 1. Introduction

Numerical computation plays an important role in the field of financial engineering, where it is essential for developing and implementing advanced mathematical models and computational techniques to solve complex financial problems [1]. As financial markets grow increasingly intricate and dynamic, numerical algorithms have become indispensable for analyzing risk, optimizing investment strategies, and pricing various financial derivatives. Numerical techniques enable the translation intricate financial models, such as option pricing models, into computationally solvable problems. These methods employ algorithms and mathematical approximations to discretize continuous equations, allowing for efficient computation on digital platforms. By harnessing the power of numerical computation, finance professionals can tackle intricate calculations and gain valuable insights into pricing, hedging, and portfolio management [2].

For instance, in option pricing, numerical techniques such as finite difference methods, Monte Carlo simulations, and numerical integration are employed to calculate option values when closed-form solutions are not attainable [3]. These methods discretize the underlying mathematical equations and allow for efficient approximation, enabling the valuation of various options, including exotic options and options with complex payoffs.

Howison [4] asserted that while the future is expected to bring forth a growing array of exotic options in the market, their modeling can still be established using the Black-Scholes model as a foundation. However, the challenge is in developing efficient and fast numerical computation for option pricing involving a very large number of options [4].

An option is a financial derivative that offers the holder the right to trade a stock or other underlying asset at a specific price in the future, known as the strike price. The holder of a call option possesses the privilege to purchase the stock or underlying asset at the strike price, whereas a put option holds the privilege to sell it. The style of options come in various forms, generally the vanilla options and exotic options. While

European and American options are considered vanilla options, Asian options fall under the category of exotic options. An Asian option is a type of path-dependent option whose payoff is determined by the average stock price over a specific time [5].

A second-order PDE can evaluate the values of the Asian options with three independent variables: time, stock price and the stock's average with a terminal condition and boundary conditions. The Black Scholes PDE for the Asian arithmetic option is [7]:

$$\frac{\partial V}{\partial t}+\frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}+S\frac{\partial V}{\partial A}-rV=0 \qquad (1)$$

Where $V$ is the value of the option, $S$ is the stock price, $A$ is the average of the stock price, $\sigma$ is the volatility of the stock price, and $r$ is the risk-free interest rate. The payoff function for a fixed strike Asian options is [8]:

$V(S,A_T,T)=max(A_T-K,\ 0)$ for the call option,

$V(S,A_T,T)=max(K-A_T,\ 0)$for the put option,

Where $T$ is the expiration time in the year, and $K$ is the strike price. The terminal condition, represented by the payoff function, is incorporated into the Black-Scholes PDE in Equation 1 to facilitate the solution process.

However, up to now, no closed-form solution is still available to solve arithmetic Asian option pricing based on Black-Scholes PDE in Equation 1 [6]. This is because the arithmetic average of a set of lognormal random variables is not lognormally distributed [7, 8]. Several transformations have been done by researchers to address the issue. Geman and Yor [9] used the Laplace transform to convert the Black-Scholes PDE in Equation 1 to an ordinary differential equation, which requires the evaluation of the inverse Laplace transform through direct numerical integration.

As such, the computational of the solution is expensive for large values of $t$. Rogers and Shi [10] introduced the scaling property and lower bound methods. The scaling property method reduces the problem to a parabolic PDE in two variables. The lower bound method considered the fact that the price of an Asian option must be greater than or equal to the price of a European call option with the same strike price and maturity. Then, it is solved numerically. Vecer [11] introduced a one-dimensional PDE for Asian option pricing by viewing the Asian option as a special case on a traded account.

Elshegmani et al. [7] solved it analytically by transforming the PDE in Equation 1 to a heat equation with constant coefficients. In addition, Elshegmani and Ahmad [8] applied the Laplace transform by transforming it into a two-dimensional Ordinary Differential Equation (ODE). Finally, they obtained an analytical solution for the fixed strike called the arithmetic Asian option.

For numerical solutions of the PDE in Equation 1, Lee and Chin [12] approximated the PDE in Equation 1 using a simple Crank-Nicolson scheme and solved using the direct method. The Crank-Nicolson scheme used was the central difference for the stock price while the backward difference for the average stock price and time, respectively [12]. This resulted in a four-point computational node on each time level and solved the linear systems using the direct method [12]. Nevertheless, Saad et al. [13] applied the standard Crank-Nicolson scheme where both the stock price and average stock price were discretized with central difference schemes, respectively, while the time level was discretized using a backward difference scheme [14]. Then, Saad et al. [13] solved the linear systems on each time level backwardly using the Gauss-Seidel (GS) iterative method.

There exists some research gap that should be addressed to enhance the numerical solutions of arithmetic Asian option pricing. Saad et al. [13] indeed utilized a higher order accuracy of the Crank-Nicolson scheme [14] compared to Lee and Chin’s work [12] to approximate the two-dimensional PDE in Equation 1. However, the study of Saad et al. [13] lacks a comprehensive stability analysis to demonstrate the scheme’s stability in the context of solving the PDE. This gap raises questions regarding the robustness and reliability of the finite difference scheme employed.

While addressing the accuracy aspect through the study of approximation schemes remains a vital endeavor in Asian option pricing research, there is an equally compelling need for the development of faster computational iterative solvers. The existing literatures has appropriately emphasized the importance of accurate Asian option valuations [6][7][8][9][10][11][12]; however, it has somewhat overlooked the computational time required to achieve such accuracy.

Although Gauss-Seidel methods have been employed in various financial modeling contexts [13][15][16][17], the application of the Red-Black strategy specifically tailored to Arithmetic Asian options pricing is notably underexplored.

RBGS is a variant of the GS method, where the main difference is that RBGS updates the variables in a checkerboard pattern, while Gauss-Seidel updates all of the variables at once. It is an iterative technique that partitions the unknowns of a linear system into red and black groups [18]. The algorithm updates the values of the unknowns iteratively, alternating between red and black groups until convergence is achieved. This alternating pattern of updates helps to accelerate the convergence of the algorithm.

Zhang [19] provides a theoretical analysis of the RBGS algorithm. The paper showed that the method converges under certain conditions and provided estimates for the convergence rate. Yavneh [20] presented numerical results for the RBGS method, and the results showed that the method is typically more efficient than other iterative methods, namely the Jacobi and the GS methods.

Thus far, the RBGS algorithm has been extensively applied by researchers for solving a wide variety of equations involving large systems of linear equations such as the 2D steady state heat conduction problem [21], Pennes bioheat equation [22], viscous forces computation in the simulation of droplet-jet collisions [23] as well as the time-dependent Ginzburg-Landau equations for anisotropic superconductors with spatially varying material properties [24]. The authors used the RBGS algorithm because it is a well-established method known to be efficient for solving large systems of equations.

In this study, the PDE in Equation 1 is approximated with the standard Crank-Nicolson approximation scheme. Then, the approximated equations are expressed in the form of linear systems at each time level. Subsequently, the RBGS algorithm is developed to solve the linear systems formed. The algorithm's performance is evaluated through comprehensive numerical experiments in terms of the number of iterations, computational time and accuracy.

## 2. Materials and Methods

### *2.1. Crank-Nicolson Approximation Scheme*

The Crank-Nicolson approximation scheme is a finite difference method that combines the explicit and implicit approaches by taking the average of the forward difference and backward difference approximations. The Black-Scholes PDE in Equation 1 can be approximated as

$$\begin{aligned}&\frac{V_{i,j,k+1}-V_{i,j,k}}{\Delta t}\\&+\frac{\sigma^2{S_i}^2}{2}\left(\frac{\begin{matrix}V_{i-1,j,k+1}-2V_{i,j,k+1}+V_{i+1,j,k+1}\\+V_{i-1,j,k}-2V_{i,j,k}+V_{i+1,j,k}\end{matrix}}{2\Delta S^2}\right)\\&+rS_i\left(\frac{V_{i+1,j,k+1}-V_{i-1,j,k+1}+V_{i+1,j,k}-V_{i-1,j,k}}{4\Delta S}\right)\\&+S_i\left(\frac{V_{i,j+1,k+1}-V_{i,j-1,k+1}+V_{i,j+1,k}-V_{i,j-1,k}}{4\Delta A}\right)\\&-r\left(\frac{V_{i,j,k+1}+V_{i,j,k}}{2}\right)=0\end{aligned}\tag{2}$$

Equation 2 can be derived and simplified as

$$\begin{aligned}a_iV_{i-1,j,k}+b_iV_{i+1,j,k}-c_iV_{i,j-1,k}+c_iV_{i,j+1,k}+V_{i,j,k}\\=F_{i,j,k+1}\end{aligned}\tag{3}$$

Where, $a_i=\frac{\alpha_i-\beta_i}{d_i}$, $b_i=\frac{\alpha_i+\beta_i}{d_i}$, $c_i=\frac{\gamma_i}{d_i}$, $F_{i,j,k+1}=\frac{f_{i,j,k+1}}{d_i}$, $\alpha_i=\frac{\Delta t\sigma^2S_i^2}{4\Delta S^2}$, $\beta_i=\frac{\Delta trS_i}{4\Delta S}$, $\gamma_i=\frac{\Delta tS_i}{4\Delta A}$, $d_i=-1-2\alpha_i+\theta$, $\theta=-\frac{\Delta tr}{2}$, $f_{i,j,k+1}=$

$$\begin{aligned}&(-\alpha_i+\beta_i)V_{i-1,j,k+1}+(-\alpha_i-\beta_i)V_{i+1,j,k+1}+\gamma_iV_{i,j-1,k+1}\\&-\gamma_iV_{i,j+1,k+1}+(-1+2\alpha_i-\theta)V_{i,j,k+1}\end{aligned}$$

and *i*, *j*, and *k* denote the nodes for stock price, *S*, average stock price, *A*, and time, *t*, respectively, on a finite difference grid.

A five-point approximation equation derived from the Crank-Nicolson approximation equation at each time level, *k*, can be illustrated in Figure. 1.

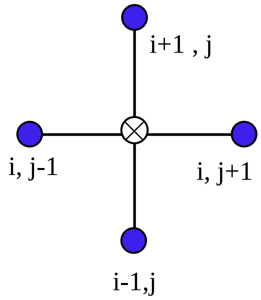

**Fig. 1 Computational nodes on each time level, *k***

### *2.2. Red-Black Gauss-Seidel Algorithm*

The approximation equations in Equation 3 will generate a sequence of linear systems on each time level of the form

$$B\underset{\sim}{V_k}=\underset{\sim}{F_{k+1}}\tag{4}$$

Where $B$ is a penta-diagonal coefficient matrix, $\underset{\sim}{F}$is the known column vector computed from the time level of $k+1$,$\underset{\sim}{V}$and is the unknown column vector for the option price at time level, *k*.

Generally, the linear system in Equation 4 can be solved using Algorithm 1 (RBGS method). The computation started from the terminal time, and the loops continued backwards in time until the time level, *k*=0.

*Algorithm 1 (RBGS Method)*

```
i.   Initializing all the parameters. Set h = 0.
ii.  Perform point iteration:
     a.  Compute red point iteration:
         for(j=1;j<=m-1;j++) {
             for (i=(j%2)+1;i<=m-1;i=i+2){
                 $V_{i,j}^{(h+1)}=F_{i,j}-a_iV_{i-1,j}^{(h+1)}-b_iV_{i+1,j}^{(h)}+c_iV_{i,j-1}^{(h+1)}-c_iV_{i,j+1}^{(h)}$
             }
         }

     b.  Compute black point iteration:
         for(j=1;j<=m-1;j++){
            for(i=2-(j%2);i<=m1;i=i+2){
$V_{i,j}^{(h+1)} = F_{i,j} - a_i V_{i-1,j}^{(h+1)} - b_i V_{i+1,j}^{(h)} + c_i V_{i,j-1}^{(h+1)} - c_i V_{i,j+1}^{(h)}$
            }
        }
iii.  Convergence test.
    a.  If the error tolerance is fulfilled, the value option at that time level is $V_{i,j}^{(h+1)}$ , and the algorithm stops.
    b.  Else, set h=h+1 and go to step ii.
```

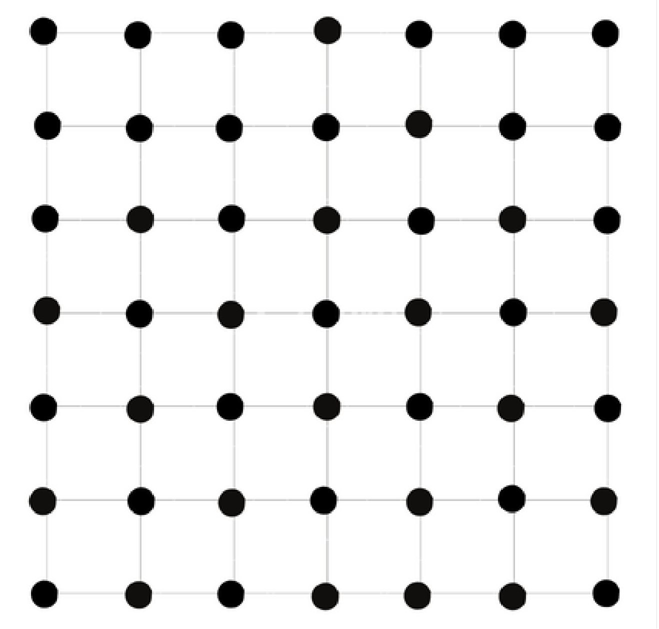

**Fig. 2 Standard computational grid**

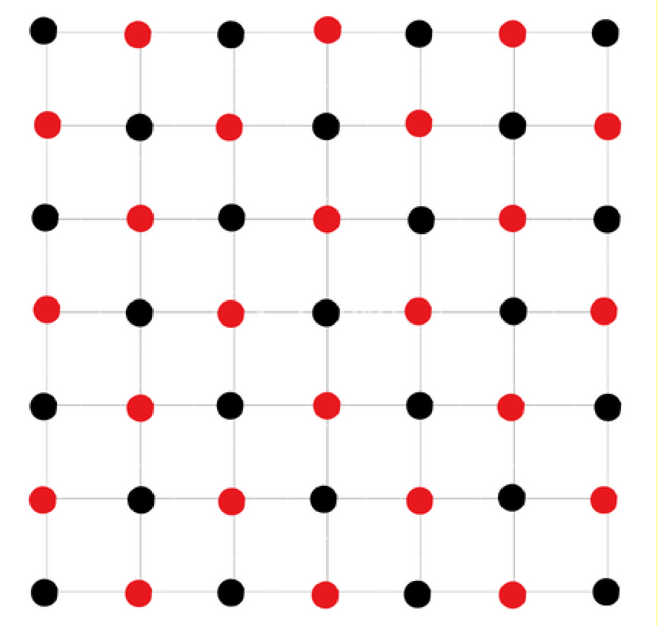

**Fig. 3 Red-Black computational grid**

On each time level, a standard computational grid is illustrated in Figure 2, where the standard GS algorithm is implemented. Nonetheless, Algorithm 1 follows the red-black ordering sequence as depicted in Figure 3. The computational node runs alternatively with red and black phases, respectively. Note that the computational of the red node depends on the black neighbouring nodes and vice versa, as shown in Figure 3 [25].

## 3. Stability Analysis

Based on the approximation difference equation in Equation 3, suppose $k+1 = N-p$ and $k = N-(p+1)$, yield:

$$\begin{aligned} &a_i V_{i-1,j,N-(p+1)} + b_i V_{i+1,j,N-(p+1)} - c_i V_{i,j-1,N-(p+1)} \\ &+c_i V_{i,j+1,N-(p+1)} + V_{i,j,N-(p+1)} \\ &= \frac{(-\alpha_i + \beta_i)}{d_i} V_{i-1,j,N-p} + \frac{(-\alpha_i - \beta_i)}{d_i} V_{i+1,j,N-p} \\ &+ \frac{\gamma_i}{d_i} V_{i,j-1,N-p} - \frac{\gamma_i}{d_i} V_{i,j+1,N-p} \\ &+ \frac{(-1+2\alpha_i-\theta)}{d_i} V_{i,j,N-p} \quad (4) \end{aligned}$$

The solutions of Equation 4 can be assumed to be the following:

$$V_{i,j,N-(p+1)} = \varepsilon^{(p+1)} e^{(i+j)2\pi\sqrt{-1}/\omega},$$

$$V_{i,j+1,N-(p+1)} = \varepsilon^{(p+1)} e^{(i+j+1)2\pi\sqrt{-1}/\omega},$$

$$V_{i,j-1,N-(p+1)} = \varepsilon^{(p+1)} e^{(i+j-1)2\pi\sqrt{-1}/\omega},$$

$$V_{i+1,j,N-(p+1)} = \varepsilon^{(p+1)} e^{(i+j+1)2\pi\sqrt{-1}/\omega},$$

$$V_{i-1,j,N-(p+1)} = \varepsilon^{(p+1)} e^{(i+j-1)2\pi\sqrt{-1}/\omega}$$

$$V_{i,j,N-p} = \varepsilon^{p} e^{(i+j)2\pi\sqrt{-1}/\omega},$$

$$V_{i,j+1,N-p} = \varepsilon^{p} e^{(i+j+1)2\pi\sqrt{-1}/\omega},$$

$$V_{i,j-1,N-p} = \varepsilon^{p} e^{(i+j-1)2\pi\sqrt{-1}/\omega},$$

$$\begin{aligned} &V_{i+1,j,N-p} = \varepsilon^{p} e^{(i+j+1)2\pi\sqrt{-1}/\omega}, \\ &V_{i-1,j,N-p} = \varepsilon^{p} e^{(i+j-1)2\pi\sqrt{-1}/\omega}. \end{aligned} \quad (5)$$

Next, by substituting Equation 5 into Equation 4, yield

$$\begin{aligned} &a_i \varepsilon^{(p+1)} e^{(i+j-1)2\pi\sqrt{-1}/\omega} + b_i \varepsilon^{(p+1)} e^{(i+j+1)2\pi\sqrt{-1}/\omega} \\ &-c_i \varepsilon^{(p+1)} e^{(i+j-1)2\pi\sqrt{-1}/\omega} + c_i \varepsilon^{(p+1)} e^{(i+j+1)2\pi\sqrt{-1}/\omega} \\ &+\varepsilon^{(p+1)} e^{(i+j)2\pi\sqrt{-1}/\omega} = \end{aligned}$$

$$\begin{aligned}
&\frac{(-\alpha_i+\beta_i)}{d_i}\varepsilon^{(p)}e^{(i+j-1)2\pi\sqrt{-1}/\omega}\\
&+\frac{(-\alpha_i-\beta_i)}{d_i}\varepsilon^{(p)}e^{(i+j+1)2\pi\sqrt{-1}/\omega}\\
&+\frac{\gamma_i}{d_i}\varepsilon^{(p)}e^{(i+j-1)2\pi\sqrt{-1}/\omega}-\frac{\gamma_i}{d_i}\varepsilon^{(p)}e^{(i+j+1)2\pi\sqrt{-1}/\omega}\\
&+\frac{(-1+2\alpha_i-\theta)}{d_i}\varepsilon^{(p)}e^{(i+j)2\pi\sqrt{-1}/\omega}
\end{aligned}$$

Then,

$$\begin{aligned}
&\varepsilon(a_ie^{-2\pi\sqrt{-1}/\omega}+b_ie^{2\pi\sqrt{-1}/\omega}\\
&-c_ie^{-2\pi\sqrt{-1}/\omega}+c_ie^{2\pi\sqrt{-1}/\omega}+1)\\
&=\frac{(-\alpha_i+\beta_i)}{d_i}e^{-2\pi\sqrt{-1}/\omega}\\
&+\frac{(-\alpha_i-\beta_i)}{d_i}e^{2\pi\sqrt{-1}/\omega}+\frac{\gamma_i}{d_i}e^{-2\pi\sqrt{-1}/\omega}\\
&-\frac{\gamma_i}{d_i}e^{2\pi\sqrt{-1}/\omega}+\frac{(-1+2\alpha_i-\theta)}{d_i}
\end{aligned}$$

$$\begin{aligned}
&\varepsilon(\frac{\alpha_i-\beta_i}{d_i}e^{-2\pi\sqrt{-1}/\omega}+\frac{\alpha_i+\beta_i}{d_i}e^{2\pi\sqrt{-1}/\omega}\\
&-\frac{\gamma_i}{d_i}e^{-2\pi\sqrt{-1}/\omega}+\frac{\gamma_i}{d_i}e^{2\pi\sqrt{-1}/\omega}+1)\\
&=\frac{(-\alpha_i+\beta_i)}{d_i}e^{-2\pi\sqrt{-1}/\omega}\\
&+\frac{(-\alpha_i-\beta_i)}{d_i}e^{2\pi\sqrt{-1}/\omega}+\frac{\gamma_i}{d_i}e^{-2\pi\sqrt{-1}/\omega}\\
&-\frac{\gamma_i}{d_i}e^{2\pi\sqrt{-1}/\omega}+\frac{(-1+2\alpha_i-\theta)}{d_i}\\
&\varepsilon[(\alpha_i-\beta_i)e^{-2\pi\sqrt{-1}/\omega}+(\alpha_i+\beta_i)e^{2\pi\sqrt{-1}/\omega}\\
&-\gamma_ie^{-2\pi\sqrt{-1}/\omega}+\gamma_ie^{2\pi\sqrt{-1}/\omega}\\
&+(-1-2\alpha_i+\theta)]\\
&=(-\alpha_i+\beta_i)e^{-2\pi\sqrt{-1}/\omega}\\
&+(-\alpha_i-\beta_i)e^{2\pi\sqrt{-1}/\omega}+\gamma_ie^{-2\pi\sqrt{-1}/\omega}\\
&-\gamma_ie^{2\pi\sqrt{-1}/\omega}+(-1+2\alpha_i-\theta)
\end{aligned}$$

$$\begin{aligned}
&\varepsilon[\alpha_i(-2+e^{-2\pi\sqrt{-1}/\omega}+e^{2\pi\sqrt{-1}/\omega})\\
&+\beta_i(e^{2\pi\sqrt{-1}/\omega}-e^{-2\pi\sqrt{-1}/\omega})\\
&+\gamma_i(e^{2\pi\sqrt{-1}/\omega}-e^{-2\pi\sqrt{-1}/\omega})\\
&-1-2\alpha_i+\theta]\\
&=\alpha_i(2-(e^{-2\pi\sqrt{-1}/\omega}+e^{2\pi\sqrt{-1}/\omega})\\
&+\beta_i(e^{-2\pi\sqrt{-1}/\omega}-e^{2\pi\sqrt{-1}/\omega})\\
&+\gamma_i(e^{-2\pi\sqrt{-1}/\omega}-e^{2\pi\sqrt{-1}/\omega})\\
&-1+2\alpha_i-\theta
\end{aligned}$$

By using identities,

$$cos(2\pi/\omega)=\frac{e^{2\pi\sqrt{-1}/\omega}+e^{-2\pi\sqrt{-1}/\omega}}{2}$$

$$sin(2\pi/\omega)=\frac{e^{2\pi\sqrt{-1}/\omega}-e^{-2\pi\sqrt{-1}/\omega}}{2\sqrt{-1}}$$

$$\begin{aligned}
&\varepsilon[\alpha_i(-2+2\,cos(2\pi/\omega))\\
&+\beta_i(2\sqrt{-1}\,sin(2\pi/\omega))\\
&+\gamma_i(2\sqrt{-1}\,sin(2\pi/\omega))\\
&-1-2\alpha_i+\theta]\\
&=\alpha_i(2-2\,cos(2\pi/\omega))\\
&+\beta_i(-2\sqrt{-1}\,sin(2\pi/\omega))\\
&+\gamma_i(-2\sqrt{-1}\,sin(2\pi/\omega))\\
&-1+2\alpha_i-\theta
\end{aligned}$$

$$\begin{aligned}
&\varepsilon[-2\alpha_i(1-cos(2\pi/\omega))\\
&+2\beta_i(\sqrt{-1}\,sin(2\pi/\omega))\\
&+2\gamma_i(\sqrt{-1}\,sin(2\pi/\omega))-1\\
&-2\alpha_i+\theta]=2\alpha_i(1-cos(2\pi/\omega))\\
&-2\beta_i(\sqrt{-1}\,sin(2\pi/\omega))\\
&-2\gamma_i\sqrt{-1}\,sin(2\pi/\omega)-1+2\alpha_i-\theta
\end{aligned}\tag{6}$$

According to von Neumann stability analysis, the difference equation is considered stable if and only if |ε|≤1. From Equation 6,

$$|\varepsilon|=\frac{\sqrt{\begin{aligned}&(2\alpha_i(1-cos(2\pi/\omega))-1+2\alpha_i-\theta)^2\\&+(sin(2\pi/\omega)(2\beta_i+2\gamma_i))^2\end{aligned}}}{\sqrt{\begin{aligned}&(-2\alpha_i(1-cos(2\pi/\omega))-1-2\alpha_i+\theta)^2\\&+(-\,sin(2\pi/\omega)(2\beta_i+2\gamma_i))^2\end{aligned}}}$$

$$|\varepsilon|=\frac{\sqrt{\begin{aligned}&([2\alpha_i(1-cos(2\pi/\omega))+2\alpha_i-\theta]-1)^2\\&+(sin(2\pi/\omega)(2\beta_i+2\gamma_i))^2\end{aligned}}}{\sqrt{\begin{aligned}&(-[2\alpha_i(1-cos(2\pi/\omega))+2\alpha_i-\theta]-1)^2\\&+(-\,sin(2\pi/\omega)(2\beta_i+2\gamma_i))^2\end{aligned}}}$$

Since $2\alpha_i(1-cos(2\pi/\omega))+2\alpha_i-\theta\geq 0$, then $|\varepsilon|\leq 1$, therefore, the approximation equation is considered to be stable. It can be concluded that the discretization scheme is unconditionally stable.

## 4. Numerical Experiments

In order to investigate the efficiency of the proposed method, several computational experiments were performed on two examples of fixed strike call arithmetic Asian option pricing. The criteria considered are the number of iterations, computational time, maximum relative error and root mean square error (RMSE) of the GS and RBGS iterative methods. The numerical results are compared with the analytical solution given by Elshegmani and Ahmad [8]. The matrix sizes tested are 100, 150, 200, 250, 300 and 350.

### *4.1. Experiment 1*

Consider the values of the parameters are $K = 20$, $r = 0.05$,

$\sigma$ = 0.25, $T$ = 1(year), $S_{max}$=$Z_{max}$=500 and $Z_0$=20 [7]. The computational results are shown in Table 1 and Figure 4.

**Table 1. Computational results of experiment 1**

| **Number of iterations** | | | | | | |
|---|---|---|---|---|---|---|
| | **Mesh Sizes** | | | | | |
| **Methods** | **100** | **150** | **200** | **250** | **300** | **350** |
| **GS** | 5480 | 10237 | 16654 | 24689 | 34319 | 45516 |
| **RBGS** | 4552 | 9082 | 15269 | 23074 | 32487 | 43464 |
| **Computational Time (Seconds)** | | | | | | |
| **GS** | 1.64 | 5.99 | 16.88 | 39.45 | 80.52 | 155.23 |
| **RBGS** | 1.10 | 4.36 | 12.98 | 30.88 | 63.65 | 137.36 |
| **Maximum Relative Errors** | | | | | | |
| **GS** | 3.1298E-8 | 3.1529E-8 | 3.1642E-8 | 3.1706E-8 | 3.1746E-8 | 3.1772E-8 |
| **RBGS** | 3.1305E-8 | 3.1539E-8 | 3.1654E-8 | 3.1721E-8 | 3.1763E-8 | 3.1790E-8 |
| | **RMSE** | | | | | |
| **GS** | 5.0497E-8 | 3.3479E-8 | 2.5037E-8 | 1.9993E-8 | 1.6639E-8 | 1.4248E-8 |
| **RBGS** | 5.0503E-8 | 3.3485E-8 | 2.5043E-8 | 2.0000E-8 | 1.6646E-8 | 1.4254E-8 |

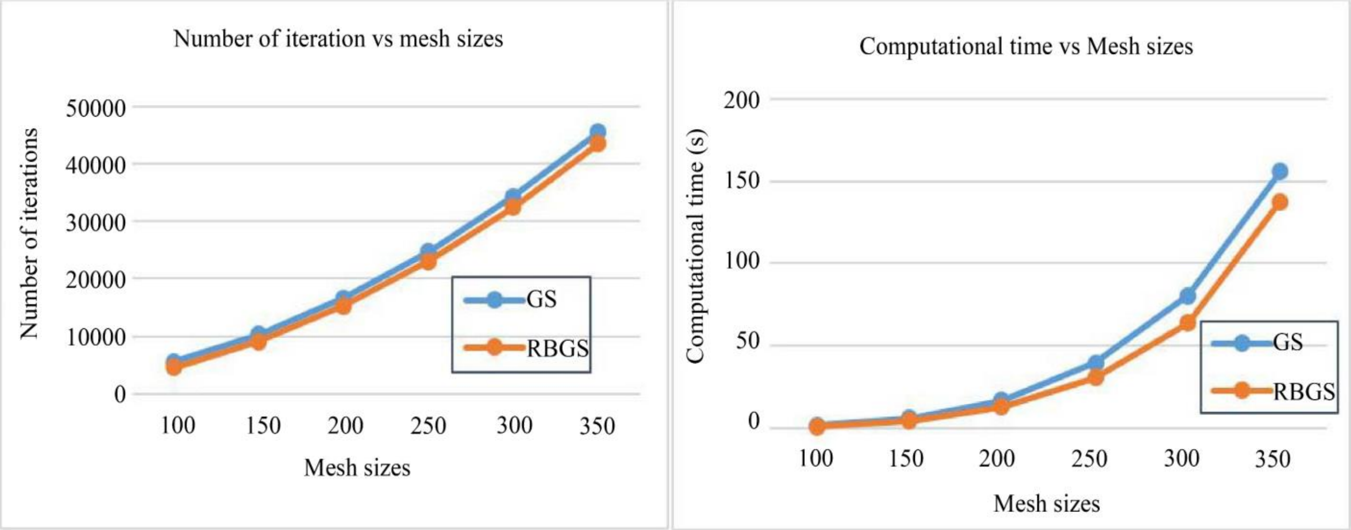

**Fig. 4 Performance of RBGS algorithm for experiment 1: (a) Comparison of the number of iterations. (b) Comparison of computational time**

According to Table 1, the RBGS method requires less number of iterations and shorter computational time than the GS method. It can be seen clearly from the graphical illustrations of the performance of the RBGS method compared to the GS method in Figure 4.

The number of iterations decreased by about 4.51% to 16.93%, while the computational time accelerated by about 11.51% to 32.93%, refer to Table 3. As for the accuracy, the maximum relative errors and RMSE shown in Table 1 for both iterative methods are in good agreement.

### *4.2. Experiment 2*

Suppose the values of the parameters are $K$ = 7, $r$ = 0.3, $\sigma$ = 0.3, $T$ = 1(year), $S_{max}$=$Z_{max}$=500, $S_0$=10 and $Z_0$=7 [12]. The computational results are tabulated in Table 2 and graphically presented in Figure 5.

Based on the computational results shown in Table 2, the RBGS algorithm again displayed a reduction in iteration count and computational time compared to the GS algorithm. Similarly, Figure 5 provides a clear visual representation of the reduction in the number of iterations and computational time achieved by the RBGS algorithm from that of the GS algorithm. The number of iterations decreased by about 3.14% to 12.33%, while the computational time accelerated by about 14.15% to 25.63%, as recorded in Table 3. Likewise, the accuracy of the RBGS algorithm is in good agreement with that of the GS algorithm, as presented at the maximum relative errors and RMSE sections in Table 2.

**Table 2. Recommended font sizes computational results of experiment 2**

| **Number of iterations** | | | | | | |
|---|---|---|---|---|---|---|
| | **Mesh Sizes** | | | | | |
| **Methods** | **100** | **150** | **200** | **250** | **300** | **350** |
| **GS** | 7083 | 13812 | 22917 | 34344 | 48052 | 64006 |
| **RBGS** | 6210 | 12700 | 21576 | 32765 | 46271 | 61998 |
| **Computational Time (Seconds)** | | | | | | |
| **GS** | 1.99 | 7.93 | 23.36 | 54.29 | 111.55 | 210.79 |
| **RBGS** | 1.48 | 6.14 | 17.41 | 41.73 | 90.50 | 180.97 |
| **Maximum Relative Errors** | | | | | | |
| **GS** | 8.7381E-7 | 8.8424E-7 | 8.8969E-7 | 8.9305E-7 | 8.9534E-7 | 8.9701E-7 |
| **RBGS** | 8.7381E-7 | 8.8424E-7 | 8.8969E-7 | 8.9305E-7 | 8.9535E-7 | 8.9701E-7 |
| | **RMSE** | | | | | |
| **GS** | 6.1139E-7 | 4.0493E-7 | 3.0270E-7 | 2.4167E-7 | 2.0112E-7 | 1.7222E-7 |
| **RBGS** | 6.1140E-7 | 4.0494E-7 | 3.0270E-7 | 2.4168E-7 | 2.0113E-7 | 1.7223E-7 |

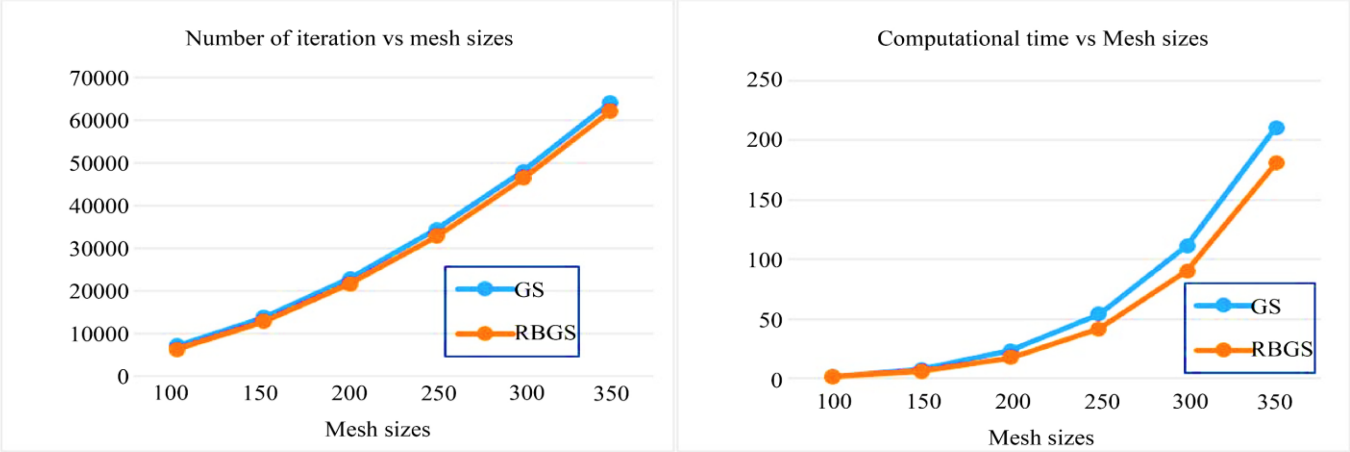

**Fig. 5 Performance of RBGS algorithm for experiment 2: (a) Comparison of the number of iterations. (b) Comparison of computational time**

**Table 3. Percentage reduction of iterations count and computational time of RBGS method relative to GS method**

| **Experiment** | **Iterations Reduced (%)** | **Time reduced (%)** |
|---|---|---|
| 1 | 4.51 – 16.93 | 11.51 – 32.93 |
| 2 | 3.14 – 12.33 | 14.15 – 25.63 |

## 5. Discussion

Based on Experiments 1 and 2, both RBGS and GS iterative methods were computed to solve the Asian option pricing with different parameters. The results in Tables 1 and 2 clearly show that the RBGS iterative method recorded less number of iterations and faster than the GS iterative method. Table 3 describes that RBGS managed to reduce the number of iterations by around 3 to 17% while accelerating the computational time by about 11 to 33%.

In terms of accuracy, both methods are quite accurate, with an accuracy of E-7 and E-8, respectively, in Experiment 1 and 2. Hence, the red-black ordering of the RBGS iterative method does not alter the accuracy of the numerical solution. The results in Table 1 and 2 show that the RMSE of the RBGS iterative method are in good agreement with the GS iterative method.

The accurate results are due to the derivation of the standard Crank-Nicolson approximation scheme onto Equation 1. Compared to the simple Crank-Nicolson approximation scheme used by Lee and Chin [12], the standard Crank-Nicolson approximation scheme applied in this paper has a higher order of accuracy. This is supported by the simulation analysis of Lee and Chin [12] and the numerical results in section 4.

The relative errors obtained by Lee and Chin [12] are about E-2 and E-3, whereas this study managed to achieve the maximum relative errors of about E-7 and E-8. Thus, the accuracy is improved by applying the standard Crank-Nicolson approximation scheme, having a central difference between stock price and average stock price while remaining the backward difference for time level.

Furthermore, the proposed RBGS method can enhance the GS method [13] efficiently regarding the number of iterations and computational time. As it is notable based on Table 3, the iteration count has been reduced to around 3 to

17%, while the computational time is accelerated by about 11 to 33%.

## 6. Conclusion

The Red-Black Gauss-Seidel (RBGS) algorithm has been developed as a means to efficiently solve the linear system arising from the Crank-Nicolson approximation equations in Equation 3 on each time level. Moreover, the Crank-Nicolson approximation equation in Equation 3 has been proven to be stable through von Neumann stability analysis. Numerical experiments have demonstrated that the RBGS algorithm outperforms the Gauss-Seidel (GS) algorithm in terms of the number of iterations and computational time required. The accuracy of the RBGS method is comparable to that of the GS method and quite accurate, as evidenced by the low RMSE values presented in Tables 1 and 2. The application of Red-Black ordering within the GS method, as exemplified by the RBGS algorithm, has shown to be an effective approach for solving arithmetic Asian option pricing. By exploiting the alternating red-black updates, the RBGS algorithm reduces the computational burden and improves the efficiency of the solution process. In conclusion, the RBGS algorithm presents a viable and efficient numerical method for solving arithmetic Asian options pricing.

## Funding Statement

This research and publication of this article were supported by research grants provided by the INTI International University Research Seeding Grant (INTI-FBC-02-02-2022) and the Office of Research Development and Consultancy (ORDC), INTI International University, respectively.

## References

[1] Abdul Quadir Md et al., “Novel Optimization Approach for Stock Price Forecasting using Multi-layered Sequential LSTM,” *Applied Soft Computing*, vol. 134, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Dan Stefanica, *A Primer for the Mathematics of Financial Engineering*, 2nd ed., United States of America: Financial Engineering Press, 2011. [Google Scholar] [Publisher Link]

[3] Ali Bolfake, Seyed Nourollah Mousavi, and Sima Mashayekhi, “Deep Learning Application in Rainbow Options,” *Advances in Mathematical Finance & Applications*, vol. 8, no. 3, pp. 951-963, 2023. [Google Scholar] [Publisher Link]

[4] S.D. Howison, “Applied Mathematics and Finance,” *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[5] John C. Hull, *Options, Futures, and Other Derivatives*, 11th Ed., New Jersey, United States of America: Pearson, 2021. [Google Scholar] [Publisher Link]

[6] Christian-Oliver Ewald, Yuexiang Wu, and Aihua Zhang, “Pricing Asian Options with Stochastic Convenience Yield and Jumps,” *Quantitative Finance*, vol. 23, no. 4, pp. 677-692, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lee Tse Yueng, “*Crank-Nicolson Scheme for Asian Option*,” M.Sc. Thesis, Universiti Tunku Abdul Rahman, Malaysia, 2012. [Google Scholar] [Publisher Link]

[8] M. Kamalakkannun, and N.D. Sridhar, “Linear Programming Based Optimal Power Flow Optimization of DCOPF for an IEEE 5 and IEEE 14 Bus System,” *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 11, pp. 95-102, 2022. [CrossRef] [Publisher Link]

[9] Hélyette Geman, and Marc Yor, “Bessel Processes, Asian Options, and Perpetuities,” *Mathematical Finance*, vol. 3, no. 4, pp. 349-375, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[10] L.C.G. Rogers, and Z. Shi, “The Value of an Asian Option,” *Journal of Applied Probability*, vol. 32, no. 4, pp. 1077-1088, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jan Vecer, “A New PDE Approach for Pricing Arithmetic Average Asian Options,” *Journal of Computational Finance*, vol. 4, no. 4, pp. 105-113, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[12] Lee Tse Yueng, and Chin Seong Tah, “A Simple Crank-Nicolson Scheme for Asian Option,” *The 6th IMT-GT Conference on Mathematics, Statistics and its Applications*, pp. 381-394, 2010. [Publisher Link]

[13] Nordin Saad, Andang Sunarto, and Azali Saudi, “Accelerated Red-Black Strategy for Image Composition using Laplacian Operator,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 1085-1195, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Steven Chapra, and Raymond Canale, *Numerical Methods for Engineers*, 8th ed., United States of America: McGraw-Hill, 2021. [Publisher Link]

[15] Okechukwu U. Solomon, “Application of Gauss-Seidel Method on Refined Financial Matrix for Solution to Partial Differential Equation (PDE) in Finance,” *Asian Journal of Pure and Applied Mathematics*, vol. 4, no. 1, pp. 666-667, 2022. [Google Scholar] [Publisher Link]

[16] Okechukwu U. Solomon, and Udoinyang I. Efiong, “A Comparative Study of Gauss-Seidel Method and Pseudo Inversion Method in Option Pricing,” *Science Set Journal of Physics*, vol. 2, no. 1, pp. 1-8, 2023. [Google Scholar] [Publisher Link]

[17] Stephen Kinsella, and Terence O'Shea, "Solution and Simulation of Large Stock Flow Consistent Monetary Production Models via the Gauss Seidel Algorithm," *Journal of Policy Modeling, Forthcoming*, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[18] Stefan Kronawitter, "*Automatic Performance Optimization of Stencil Codes*," Doctoral Dissertation, Universität Passau, Germany, 2019. [Google Scholar] [Publisher Link]

[19] Jun Zhang, "Acceleration of Five-Point Red-Black Gauss-Seidel in Multigrid for Poisson Equation," *Applied Mathematics and Computation*, vol. 80, no. 1, pp. 73-93, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[20] Irad Yavneh, "Multigrid Smoothing Factors for Red-Black Gauss–Seidel Relaxation Applied to a Class of Elliptic Operators," *SIAM Journal on Numerical Analysis*, vol. 32, no. 4, pp. 1126-1138, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[21] Mahmoud El Maghrbay, Reda Ammar, and Sanguthevar Rajasekaran, "Fast GPU Algorithms for Implementing the Red-Black Gauss-Seidel Method for Solving Partial Differential Equations," *IEEE Symposium on Computers and Communications*, pp. 269-274, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[22] Cosmo D. Santiago et al., "A Multigrid Waveform Relaxation Method for Solving the Pennes Bioheat Equation," *Numerical Heat Transfer, Part A: Applications*, vol. 83, no. 9, pp. 976-990, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Johanna Potyka et al., "Towards DNS of Droplet-Jet Collisions of Immiscible Liquids with FS3D," *arXiv*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] C.W.W. Haddon, and D.P. Hampshire, "Fast Multigrid Simulations of Pinning in REBCO with Highly Resistive Nanorods," *IEEE Transactions on Applied Superconductivity*, vol. 33, no. 5, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]