

Article

# Finding an Efficient Computational Solution for the Bates Partial Integro-Differential Equation Utilizing the RBF-FD Scheme

Gholamreza Farahmand <sup>1</sup>, Taher Lotfi <sup>1,\*</sup>, Malik Zaka Ullah <sup>2</sup> and Stanford Shateyi <sup>3,\*</sup><sup>1</sup> Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran<sup>2</sup> Mathematical Modeling and Applied Computation (MMAC) Research Group, Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia<sup>3</sup> Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, P. Bag X5050, Thohoyandou 0950, South Africa

\* Correspondence: lotfi@iauh.ac.ir (T.L.); stanford.shateyi@univen.ac.za (S.S.)

**Abstract:** This paper proposes a computational solver via the localized radial basis function finite difference (RBF-FD) scheme and the use of graded meshes for solving the time-dependent Bates partial integro-differential equation (PIDE) arising in computational finance. In order to avoid facing a large system of discretization systems, we employ graded meshes along both of the spatial variables, which results in constructing a set of ordinary differential equations (ODEs) of lower sizes. Moreover, an explicit time integrator is used because it can bypass the need to solve the large discretized linear systems in each time level. The stability of the numerical method is discussed in detail based on the eigenvalues of the system matrix. Finally, numerical tests revealed the accuracy and reliability of the presented solver.

**Keywords:** PIDE; stochastic volatility; semi-discretization; RBF-FD; Bates model

**MSC:** 65M22; 91G60; 91B25



**Citation:** Farahmand, G.; Lotfi, T.; Ullah, M.Z.; Shateyi, S. Finding an Efficient Computational Solution for the Bates Partial Integro-Differential Equation Utilizing the RBF-FD Scheme. *Mathematics* **2023**, *11*, 1123. <https://doi.org/10.3390/math11051123>

Academic Editors: Maria Isabel Berenguer and Manuel Ruiz Galán

Received: 18 January 2023

Revised: 3 February 2023

Accepted: 21 February 2023

Published: 23 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introductory Notes

The Bates model for option pricing considers that the underlying asset  $S_t$ , the volatility  $V_t$ , the riskless constant  $r$  and  $N_t$  as the Poisson process satisfy the following system of stochastic differential equations (SDEs) [1]:

$$\begin{aligned}dS_t &= \sqrt{V_t} S_t dW_t^1 + (-\lambda \xi - q + r) S_t dt + (q - 1) S_t dN_t, \\dV_t &= \sigma \sqrt{V_t} dW_t^2 + \kappa(-V_t + \theta) dt,\end{aligned}\tag{1}$$

wherein  $W_t^2$  and  $W_t^1$  are standard Brownian motions having  $dW_t^1 dW_t^2 = \rho dt$ . Here  $\kappa$  is the reversion's rate of the variance  $V_t$ ,  $\lambda$  is the Poisson process intensity,  $\xi$  is the mean jump,  $q$  is the dividend,  $\rho$  is the jump size, while  $\theta$  is the mean level and  $\sigma$  stands for the volatility fixed value.

Financial derivatives such as European call or put options play pioneer roles in the risk management of some portfolios and their pricing as efficiently as possible is of importance. On the other hand for the financial derivative price, since analytical relations are available only in limited settings, one is in need for the construction and the application of fast and stable numerical solvers. More concretely, starting from the initial time zero, we must numerically solve a second-order high-dimensional time-dependent partial integro-differential equation (PIDE) or a partial differential equation (PDE) and then compute the present value of the financial derivative [2–4].

The Heston model, which could be considered as a generalization of the Black–Scholes model [5], can be extended further if one follows the consideration of Bates [1,6] by imposing the jump component into the modeling. In fact, in the stochastic volatility jump (SVJ) model, the price of an option is computed by solving a time-dependent 2D PIDE [7,8]. It is requisite to recall some related models [9,10] discussing stochastic volatility for PDEs in control theory and AI.

The Bates PIDE based on the price function  $u(x, y, \tau)$  for European options is expressed by the following [11]:

$$\begin{aligned} \frac{\partial u(x, y, \tau)}{\partial \tau} &= \frac{1}{2}yx^2 \frac{\partial^2 u(x, y, \tau)}{\partial x^2} + \frac{1}{2}\sigma^2y \frac{\partial^2 u(x, y, \tau)}{\partial y^2} \\ &+ \rho\sigma yx \frac{\partial^2 u(x, y, \tau)}{\partial x\partial y} \\ &+ (-\lambda\xi - q + r)x \frac{\partial u(x, y, \tau)}{\partial x} + \kappa(\theta - y) \frac{\partial u(x, y, \tau)}{\partial y} \\ &- (\lambda + r)u(x, y, \tau) + \lambda \int_0^\infty u(x\varrho, y, \tau)b(\varrho)d\varrho = \mathcal{A}u(x, y, \tau), \end{aligned} \tag{2}$$

wherein  $T$  is the time to maturity and  $\tau = T - t$  is a time transformation to have forward in time PIDE formulation, unlike the original Bates PIDE, which is backward in time. Besides, both the differential and integral parts of the PIDE problem have been encapsulated in the operator  $\mathcal{A}$ , that is to say, we also can write

$$\mathcal{A}u(x, y, \tau) = \mathcal{A}_D u(x, y, \tau) + \lambda \mathcal{A}_I u(x, y, \tau), \tag{3}$$

in which  $\mathcal{A}_D$  and  $\mathcal{A}_I$  stand for the differential and integral portions of the PIDE problem. The probability density function is  $b(\varrho) = \frac{1}{\sqrt{2\pi\hat{\sigma}\varrho}} \exp\left[-\frac{(\ln(\varrho)-\gamma)^2}{2\hat{\sigma}^2}\right]$ , where it reads  $\int_0^\infty b(\varrho)d\varrho = 1$ . Here,  $\hat{\sigma}$  and  $\gamma$  are the standard deviation and the mean, respectively, which are positive constants. Additionally, we have  $\xi = \exp\left(\gamma + \frac{1}{2}\hat{\sigma}^2\right) - 1$ .

The so-called payoff which is the the initial condition for the PIDE problem in call-type option pricing can be expressed as [12]:

$$u(x, y, 0) = (0, x - K)^+, \tag{4}$$

wherein  $K$  is the strike price. The payoff for a put option could be written similarly. The point is that the initial condition is written only on  $x$  and does not rely on the second independent variable of the PIDE, i.e.,  $y$ .

The side conditions for  $x$  and  $y$  could be given as follows [12]:

$$u(x, y, \tau) \simeq 0, \quad x \rightarrow 0, \tag{5}$$

$$u(x, y, \tau) \simeq x_{\max} \exp(-q\tau) - K \exp(-r\tau), \quad x \rightarrow x_{\max}, \tag{6}$$

$$\frac{\partial u(x, y, \tau)}{\partial y} \simeq 0, \quad y \rightarrow +y_{\max}. \tag{7}$$

Note that for the case when  $y = 0$ , the PIDE (2) is degenerate and no boundaries should be incorporated while  $x_{\max}$  and  $y_{\max}$  are large constants. Similarly for the put option, the boundary conditions are described by the following:

$$u(x, y, \tau) \simeq K \exp(-r\tau) - x_{\max} \exp(-q\tau), \quad x \rightarrow 0, \tag{8}$$

$$u(x, y, \tau) \simeq 0, \quad x \rightarrow x_{\max}, \tag{9}$$

$$\frac{\partial u(x, y, \tau)}{\partial y} \simeq 0, \quad y \rightarrow y_{\max}. \tag{10}$$

The Bates PIDE (2) is given on  $(x, y, \tau) \in [0, +\infty) \times [0, +\infty) \times (0, T]$ . To solve our high-dimensional linear PDE, we must truncate the unbounded domain while quite delicately ignoring the error caused by imposing the boundary conditions. This can be pursued as follows:

$$\Omega = [0, x_{\max}] \times [0, y_{\max}], \tag{11}$$

wherein  $x_{\max}, y_{\max}$  are fixed values. The values for  $x_{\max}$ , and  $y_{\max}$  should be considered large enough to be able to neglect the effect of imposing artificial boundary conditions or imposing the boundary conditions for truncated domains. Some choices are  $\Omega = [0, 4K] \times [0, 1]$  or  $\Omega = [0, 3K] \times [0, 1]$ .

Assume that  $\{x_i\}_{i=1}^m$  is a mesh of nodes for  $x$ . The hyperbolic stretching of nodes [13] can be expressed as follows ( $1 \leq i \leq m$ ):

$$x_i = c \sinh(\beta_i) + K, \tag{12}$$

wherein  $c > 0$  stands for a fixed value that controls the density around  $x = K$  and  $m \gg 3$ . In implementations, one can employ  $c$  as in [14], i.e.,  $c = K/5$ . This puts a focus around the strike price, in which the initial condition of the PIDE has nonsmoothness. Moreover,  $\{\beta_i\}_{i=0}^m$  stands for the uniform points given by the following:

$$\beta_i = (i - 1)\Delta\beta + \sinh^{-1}\left(-\frac{K}{c}\right), \quad 1 \leq i \leq m, \tag{13}$$

wherein  $\Delta\beta = (m - 1)^{-1} \left( \sinh^{-1}\left(\frac{S-K}{c}\right) - \sinh^{-1}\left(-\frac{K}{c}\right) \right)$ .

Also, if  $\{y_j\}_{j=1}^n$  is a partition for  $y$ , then this stretching strategy can be expressed by the following:

$$y_j = \sinh(\zeta_j)v, \quad 1 \leq j \leq n, \tag{14}$$

wherein  $v > 0$  is a fixed value that controls the density around  $y = 0$  and  $n \gg 3$ . Basically, we use  $v = K/500$  [14]. Additionally, the  $\zeta_j$  are equidistant nodes provided by  $\zeta_j = (\Delta\zeta)(j - 1)$ , and for any  $1 \leq j \leq n$  we have  $\Delta\zeta = \frac{1}{n-1} \sinh^{-1}\left(\frac{K}{v}\right)$ .

Numerical solution methods generally utilize the discretization means to realize the approximate calculation. When the computational domain/interval is partitioned more finely, the calculated result is closer to the theoretical solution. Indeed, the time required for the calculation increases. For high-dimensional PIDE problems with kink behavior at the initial conditions, sometimes special solvers such as high-order sparse numerical methods are necessary, see, e.g., [15]. Noting finite difference (FD) methods are discussed in [16,17].

In this paper, the main aim is to propose a novel computational method for resolving (2) via the radial basis function generated finite difference (RBF-FD) methodology [18]. This is mostly because (2) is a (1+2)D problem with variable coefficients, in which there is one cross derivatives. Hence, the computational solvers should be constructed for this aim with much attention. In fact, the motivation of this work lies in the fact that literature lacks the application of efficient RBF-FD methodology which result in fast and sparse procedures for solving the Bates PIDE model. Hence, such an application and investigation on the theoretical stability issues will help price option under stochastic volatility in equity markets.

The RBF-FD formulations in this paper, see, e.g., [19], are written so they can be applied to graded meshes in which there is a clear concentration on the hot zone. The procedure taken here is to employ tensor grids and then time discretize the semi-discretized constructed problem. We note that the present work is related to the pioneering works in [20–22]. Meanwhile, these works motivate us to propose a new variant of the RBF-FD scheme for the Bates PIDE problem that competes with these efficient works.

In this paper, after reviewing the well-resulted maps for generating graded meshes along spatial variables with a clear focus around the hot area, the rest of this article is unfolded as follows. The RBF-FD formulas associated with the GMQ RBF are given in

Section 2. Then, the semi-discretization of the two-dimensional PDE (2) is described in Section 3. Then in Section 4, an explicitly quadratically convergent method is taken into account. In fact, an explicit time integrator is used because it can avoid to the need to solve the large discretized linear systems in each time level. It is shown that the proposed solver is fast and conditionally stable. The numerical pieces of evidence are given in Section 5, which overwhelmingly uphold the theoretical discussions of the paper. Concluding notes are provided in Section 6.

### 2. RBF-FD: The Weights

Generally speaking, for computing the weights  $\alpha_i$  in the methodology of the RBF-FD, one must consider  $L$  as a linear operator and at  $\underline{x} = \underline{x}_p$ , for the node locations  $\underline{x}_i$ , the following is written down [20]:

$$\begin{bmatrix} \Lambda_1(\underline{x}_1) & \Lambda_1(\underline{x}_2) & \cdots & \Lambda_1(\underline{x}_m) \\ \Lambda_2(\underline{x}_1) & \Lambda_2(\underline{x}_2) & \cdots & \Lambda_2(\underline{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_m(\underline{x}_1) & \Lambda_m(\underline{x}_2) & \cdots & \Lambda_m(\underline{x}_m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} L\Lambda_1(\underline{x})|_{\underline{x}=\underline{x}_p} \\ L\Lambda_2(\underline{x})|_{\underline{x}=\underline{x}_p} \\ \vdots \\ L\Lambda_m(\underline{x})|_{\underline{x}=\underline{x}_p} \end{bmatrix}, \tag{15}$$

where the underlined  $\underline{x}$  shows a vector quantity in the dimension  $d$  and  $1 \leq k \leq m$  for some set of test functions  $\Lambda(\underline{x})$ . It is noted that the extension of RBF-FD methodology for solving computational finance models was revived by the works of Soleymani and co-authors, see for instance [23,24].

Now, we consider the famous generalized multiquadric RBF (GMQ RBF) as follows ([25] Chapter 4):

$$\Lambda(r_i) = (p^2 + r_i^2)^l, \quad i = 1, 2, \dots, m, \tag{16}$$

where  $l$  is a suitable parameter, the parameter of shape is  $p$  and  $r_i = \|\mathbf{y} - \mathbf{y}_i\|$  shows the Euclidean distance.

It is now focused on computing the weights for the GMQ RBF (in the 1D case without loss of generality). So, we consider a graded mesh including three points along the first spatial variable. For finding the weights of the RBF-FD methodology, by taking into account  $L$  as an operator, we could write down [26]:

$$L[\Lambda(y_j)] \simeq \sum_{i=1}^{\psi} \alpha_i \Lambda(y_i), \quad j = 1, 2, \dots, \psi. \tag{17}$$

This gives us  $\psi$  unknowns for  $\psi$  equations while the solutions will be  $\alpha_i$ . For computing the 1st derivative, three graded nodes are considered ( $\psi = 3$ ) as comes next:  $\{y_i - h, y_i, y_i + wh\}$ ,  $w > 0$ ,  $h > 0$ , and find (17) as follows:

$$g'(y_i) \simeq \alpha_{i-1}g(y_{i-1}) + \alpha_i g(y_i) + \alpha_{i+1}g(y_{i+1}). \tag{18}$$

Noting that we assume that the function  $g$  is smooth sufficiently. In estimating the 1st derivative of a function, the analytical weighting coefficients associated to this RBF can be given as follows [22]:

$$\alpha_{i-1} = \frac{\omega(p^2(9 - 6l) - h^2(l - 1)(4(l - 5)\omega - 10l + 29))}{3p^2h(2l - 3)(\omega + 1)}, \tag{19}$$

$$\alpha_i = \frac{(\omega - 1)(p^2(6l - 9) + 4h^2(l - 5)(l - 1)\omega)}{3p^2h(2l - 3)\omega}, \tag{20}$$

$$\alpha_{i+1} = \frac{p^2(6l - 9) - h^2(l - 1)\omega(2l(5\omega - 2) - 29\omega + 20)}{3p^2h(2l - 3)\omega(\omega + 1)}. \tag{21}$$

Similarly, in estimating the function’s second derivative, we can obtain

$$g''(y_i) \simeq \sum_{j=i-1}^{i+1} \Theta_j g(y_j), \tag{22}$$

along with the following weighting coefficients:

$$\Theta_{i-1} = \frac{2(p^2(6l - 9) - h^2(l - 1)(4(l - 5)\omega^2 + (34 - 8l)\omega + 10l - 29))}{3p^2h^2(2l - 3)(\omega + 1)}, \tag{23}$$

$$\Theta_i = \frac{2(p^2(9 - 6l) + h^2(l - 1)(4(l - 5)\omega^2 + (25 - 2l)\omega + 4(l - 5)))}{3p^2h^2(2l - 3)\omega}, \tag{24}$$

$$\Theta_{i+1} = \frac{2(p^2(6l - 9) - h^2(l - 1)(2l(\omega(5\omega - 4) + 2) + \omega(34 - 29\omega) - 20))}{3p^2h^2(2l - 3)\omega(\omega + 1)}. \tag{25}$$

Also noting that the given RBF-FD formulations are valid for the interior nodes and at boundary points, similar formulations must be constructed. We give the derivation for the independent variable  $y$  and it would be similar for the other cases. The formulations (19)–(21) and (23)–(24) are useful for the rows two to the row before the last one, while for the 1st and the last rows of the derivative matrices (30) and (31), the weighting coefficients could not be valid on boundaries and sided estimations should be incorporated. Hence, by the work [21] on the stencil  $\{y_1, y_2, y_3\}$ , we have:

$$g'(y_1) = g[y_2, y_1] - g[y_3, y_2] + g[y_3, y_1] + \mathcal{O}\left((y_2 - y_1)^2\right), \tag{26}$$

and

$$g'(y_m) = -g[y_{m-1}, y_{m-2}] + g[y_{m-2}, y_m] + g[y_m, y_{m-1}] + \mathcal{O}\left((y_{m-1} - y_m)^2\right), \tag{27}$$

wherein  $g[l, p] = (g(l) - g(p)) / (l - p)$ . In a similar manner, for the four nodes  $\{y_1, g(y_1)\}, \{y_2, g(y_2)\}, \{y_3, g(y_3)\}, \{y_4, g(y_4)\}$ , we can obtain

$$g''(y_1) = \frac{2(\delta y_{1,2} + \delta y_{1,3} + \delta y_{1,4})}{\delta y_{1,2}\delta y_{1,3}\delta y_{1,4}} g(y_1) + \frac{2(\delta y_{3,1} + \delta y_{4,1})}{\delta y_{1,2}\delta y_{2,3}\delta y_{2,4}} g(y_2) + \frac{2(\delta y_{2,1} + \delta y_{4,1})}{\delta y_{1,3}\delta y_{3,2}\delta y_{3,4}} g(y_3) + \frac{2(\delta y_{2,1} + \delta y_{3,1})}{\delta y_{1,4}\delta y_{4,2}\delta y_{4,3}} g(y_4) + \mathcal{O}\left(h^2\right), \tag{28}$$

where  $\delta y_{l,q} = y_l - y_q$ ,  $h$  is the maximum space width for the considered stencil nodes. Similarly, we have:

$$g''(y_m) = \frac{2(\delta y_{m-3,m} + \delta y_{m-2,m} + \delta y_{m-1,m})}{\delta y_{m-3,m}\delta y_{m,m-2}\delta y_{m,m-1}} g(y_m) + \frac{2(\delta y_{m-3,m} + \delta y_{m-2,m})}{\delta y_{m-3,m-1}\delta y_{m-1,m-2}\delta y_{m-1,m}} g(y_{m-1}) + \frac{2(\delta y_{m-3,m} + \delta y_{m-1,m})}{\delta y_{m-3,m-2}\delta y_{m-2,m-1}\delta y_{m-2,m}} g(y_{m-2}) + \frac{2(\delta y_{m-2,m} + \delta y_{m-1,m})}{\delta y_{m-2,m-3}\delta y_{m-1,m-3}\delta y_{m,m-3}} g(y_{m-3}) + \mathcal{O}\left(h^2\right). \tag{29}$$

### 3. A New Solution Method

Let us use the well-known procedure of method of lines (MOL) for semi discretization of the time-dependent problem [27,28] and convert the PIDE problem into a set of linear ordinary differential equations (ODEs). Hence, the following derivative matrices for the

1st and 2nd derivatives of the function in approximating the PDE problem (2) via semi-discretization are considered on non-uniform stencils given in Section 2 as comes next:

$$M_x = \begin{cases} \alpha_{i,j} \text{ using (19)} & i - j = 1, \\ \alpha_{i,j} \text{ using (20)} & i - j = 0, \\ \alpha_{i,j} \text{ using (21)} & j - i = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

and

$$M_{xx} = \begin{cases} \Theta_{i,j} \text{ using (23)} & i - j = 1, \\ \Theta_{i,j} \text{ using (24)} & i - j = 0, \\ \Theta_{i,j} \text{ using (25)} & j - i = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

Consider the  $N \times N$  unit matrix  $I = I_x \otimes I_y$ , while  $N = m \times n$ ,  $I_x$  and  $I_y$  are unit matrices of appropriate sizes. The MOL can be resulted in the following coefficient matrix for the 1 + 2 dimensional PIDE:

$$B = \frac{1}{2} \mathcal{Y} \mathcal{X}^2 (M_{xx} \otimes I_n) + \frac{1}{2} \sigma^2 \mathcal{Y} (I_m \otimes M_{yy}) + \rho \sigma \mathcal{Y} \mathcal{X} M_{x,y} + (-\lambda \xi - q + r) \mathcal{X} (M_x \otimes I_n) + \kappa (\theta I_N - \mathcal{Y}) (I_m \otimes M_y) - (-\lambda + r) I_N, \tag{32}$$

where  $\otimes$  stands for the Kronecker product. The square matrices  $M_x, M_y, M_z, M_{xx}$ , and  $M_{yy}$ , are constructed by the associated weights similarly. Additionally, the sparse diagonal matrices  $\mathcal{Y}$  and  $\mathcal{X}$  are written as:

$$\mathcal{Y} = I_x \otimes \text{diag}(y_1, y_2, \dots, y_n), \tag{33}$$

$$\mathcal{X} = \text{diag}(x_1, x_2, \dots, x_m) \otimes I_y. \tag{34}$$

Here the weights corresponding the cross derivative term in the structure of the PIDE (2) can be obtained by employing the Kronecker product as follows:

$$M_{x,y} = M_x \otimes M_y. \tag{35}$$

Now it is possible to find the following system of ODEs for pricing (2):

$$u'(\tau) = Bu(\tau). \tag{36}$$

Now, note that we can use the work of [22,29] to discretize the integral part as follows. By a linear interpolation for  $u(x\varrho, y, \tau)$  among the adaptive numerical grid nodes, the nonlocal integral given in (2) can be solved using

$$\mathcal{A}_I(u) = \int_0^\infty u(x\varrho, y, \tau) b(\varrho) d\varrho. \tag{37}$$

Employing  $z = x\varrho$ , one can transform (37) into the integral below:

$$\mathcal{A}_I(u) = \int_0^\infty u(z, y, \tau) b\left(\frac{z}{x}\right) \left(\frac{1}{x}\right) dz. \tag{38}$$

Using a linear interpolation for (38), we can find the following:

$$\mathcal{A}_i(u) \simeq \sum_{l=1}^{m-1} Q_{i,l}, \tag{39}$$

for every node  $x_i, i = 2, \dots, m - 1$ , wherein

$$Q_{i,l} = \int_{x_l}^{x_{l+1}} \left( \frac{x_{l+1} - z}{\Delta x_l} u(x_l, y, \tau) + \frac{z - x_l}{\Delta x_l} u(x_{l+1}, y, \tau) \right) b \left( \frac{z}{x_i} \right) \left( \frac{1}{x_i} \right) dz, \tag{40}$$

wherein  $\Delta x_l = x_{l+1} - x_l$  is the graded step size. Hence, we have

$$Q_{i,l} = \frac{1}{2\Delta x_l} \left( \exp \left( \gamma + \frac{\hat{\sigma}^2}{2} \right) \left( -\operatorname{erf} \left( \frac{-\ln \left( \frac{x_l}{x_i} \right) + \gamma + \hat{\sigma}^2}{\sqrt{2}\hat{\sigma}} \right) \right. \right. \\ \left. \left. + \operatorname{erf} \left( \frac{-\ln \left( \frac{x_{l+1}}{x_i} \right) + \gamma + \hat{\sigma}^2}{\sqrt{2}\hat{\sigma}} \right) \right) x_i (u(x_l, y, \tau) - u(x_{l+1}, y, \tau)) + \left( \operatorname{erf} \left( \frac{\gamma - \ln \left( \frac{x_l}{x_i} \right)}{\sqrt{2}\hat{\sigma}} \right) \right. \right. \\ \left. \left. - \operatorname{erf} \left( \frac{\gamma - \ln \left( \frac{x_{l+1}}{x_i} \right)}{\sqrt{2}\hat{\sigma}} \right) \right) (x_{l+1} u(x_l, y, \tau) - x_l u(x_{l+1}, y, \tau)) \right), \tag{41}$$

wherein  $\operatorname{erf}(\cdot)$  stands for the Gaussian distribution.

So, (36) turns into

$$u'(\tau) = \bar{B}u(\tau), \tag{42}$$

where  $\bar{B}$  is the system matrix after imposing the integral part. Finally, after considering the boundary conditions, a set of ODEs can be attained as follows:

$$u'(\tau) = F(\tau, u(\tau)) = \bar{B}u(\tau) + b, \tag{43}$$

wherein  $b$  consists of the boundary conditions.

#### 4. The Time-Stepping Solver

Time stepping schemes must be used to solve (43). Although very recently some optimal time stepping solvers have been proposed in literature [30–32] for solving system of ODEs, here we focus on a basic but efficient one. Now it is considered that  $\mathbf{u}^t$  as an approximation to  $u(\tau_t)$ , then we could derive our final (explicit) time-integrator method. Select  $k + 1$  uniform temporal nodes and  $0 \leq \iota \leq k, \tau_{\iota+1} = \tau_\iota + \zeta, \zeta = \frac{T}{k} > 0$  with  $\mathbf{u}^0 = (4)$ , then the second-order RK solver (RK2) also known as the mid-point explicit method is given by [33] (p. 95):

$$\mathbf{u}^{\iota+1} = \mathbf{u}^\iota + \psi_2 + \mathcal{O}(\zeta^3), \tag{44}$$

where

$$\psi_2 = \zeta F \left( \tau_\iota + \frac{1}{2}\zeta, \mathbf{u}^\iota + \frac{1}{2}\psi_1 \right), \quad \psi_1 = \zeta F(\tau_\iota, \mathbf{u}^\iota). \tag{45}$$

The approach (44) is useful because its explicit procedure helps programming in lower computational time than many of its competitors from the RK methods. This is a motivation of choosing (44) and not other higher order members of the RK family since their computational cost per time level increases. Anyhow, the investigation for finding the best time-stepping solver from the RK family of an optimal order for our specific PIDE problem remains an open question which could be focused on forthcoming works. Now the most important thing is to investigate that under what conditions this stability can be kept.

**Theorem 1.** *Let us assume that (43) satisfies the Lipschitz condition, then we have a conditional time-stable iteration process using (44) for solving (43).*



**Proof.** Considering the time-stepping method (44) on the set of ODEs (43) gives:

$$\mathbf{u}^{t+1} = \left( \frac{(\zeta \bar{B})^0}{1} + \frac{(\zeta \bar{B})^1}{1} + \frac{(\zeta \bar{B})^2}{2} \right) \mathbf{u}^t. \tag{46}$$

The explicit method (46) is clearly time-stable if the matrix eigenvalues of

$$\left( I + \zeta \bar{B} + \frac{(\zeta \bar{B})^2}{2} \right) \tag{47}$$

have modulus less than or equal to one. Viewing (46) as an iterative map, it would be clear that the eigenvalues of this matrix are  $1 + \zeta \bar{B}_i + \frac{(\zeta \bar{B}_i)^2}{2}$ , where  $\bar{B}_i$  are the eigenvalues of matrix  $\bar{B}$ . Thus, for  $i = 1, 2, \dots, m$ , the A-stability is simplified to

$$\left| 1 + \zeta \bar{B}_i + \frac{(\zeta \bar{B}_i)^2}{2} \right| \leq 1. \tag{48}$$

Therefore, our proposed method is time-stable if the time step size  $\zeta$  reads as (48). The stability function in (48) shows a conditional stable behavior for (44). Using (48) along with  $\zeta > 0$  we have the following:

$$0 < \zeta \leq \frac{2}{\text{Re}(\lambda_{\max}(\bar{B}))}, \tag{49}$$

where  $\text{Re}(\cdot)$  is the real part and  $\lambda_{\max}(\cdot)$  is the largest eigenvalue (in the absolute value sense). Note that we also obtain

$$-\zeta_i \leq \text{Im}(\bar{B}_i) \leq \zeta_i, \tag{50}$$

while

$$\zeta_i = \left( 2 \left( -\frac{\text{Re}(\bar{B}_i)(\zeta \text{Re}(\bar{B}_i) + 2)}{\zeta^3} \right)^{1/2} - \frac{\text{Re}(\bar{B}_i)(\zeta \text{Re}(\bar{B}_i) + 2)}{\zeta} \right)^{1/2}. \tag{51}$$

These inequalities on the real and the imaginary parts of the eigenvalues will determine the conditional time stability bounds of the proposed solver when pricing (2). This ends the proof.  $\square$

To discuss about the advantages of the proposed approach, we briefly express that our solver has now been expressed all in matrix notations as in (43) which is a system of linear ODEs. When it couples by the ODE solver (44) with the stability condition (50), it solves (2) and the stability relied only on the largest eigenvalue of the system matrix.

### 5. Numerical Aspects

The goal here is to resolve (2) for at-the-money options, i.e., the value of  $u$  at  $x_0 = K$  and also  $y_0 = 0.04$  and  $K = 100\$$ . The comparing methods are given below:

- The 2nd-order FD scheme with equidistant stencils for space and the explicit 1st order Euler’s scheme denoted by FDM,
- The method of scalable algebraic multigrid discussed in [34] and shown by AFFT.
- The scheme recently proposed by Soleymani et al. in [21] based on efficient non-uniform procedure denoted by SM.
- The presented solver in Sections 2–4 shown via RBF-FD-PM in this section.

Noting that all the programs have been written carefully under similar conditions in Mathematica 13 [35,36]. Here, the whole CPU time (for constructing the meshes, the derivative matrices, the set of ODEs and employing the time-stepping method) is in second.



We use more number of nodes along  $x$  than  $y$ , since its computational domain is larger. The criterion given below is used for computing the errors

$$\varepsilon_{i,j,t} = |u_{\text{approx}}(x_i, y_j, \tau_i) - u_{\text{ref}}(x, y, \tau)|, \tag{52}$$

where  $u_{\text{approx}}$  and  $u_{\text{ref}}$  are the approximate and exact solutions.  $u_{\text{ref}}$  is selected from the already well-known literature [14,34].

It is remarked that one efficient way to compute the shape parameter is to calculate it adaptively via the number of discretization points, the numerical domain as well as the structure of the PIDE problem. Hence, here we use  $(1 \leq i \leq m - 1)$ :

$$p = 4 \max\{\Delta x_i\}, \tag{53}$$

where  $\Delta x_i$  are the increments along the variable mesh. We can write and use (53) similarly for the other variable. Throughout the tables of this paper, a E-b stands for the scientific notation  $a \times 10^{-b}$ .

**Example 1 ([14]).** Let us investigate the computational results for the call option of (2) using the following settings:

$$\rho = -0.9, r = 0.025, \lambda = 0, \sigma = 0.3, \kappa = 1.5, \theta = 0.04, q = 0, T = 1. \tag{54}$$

The reference price, which is obtained by the FFT approach [14], is 8.894869 at the point  $(x_0, y_0) = (100, 0.04)$ . The numerical truncated domain is  $\Omega = [0, 3K] \times [0, 1]$  and  $\psi = 1.5$ . Economically speaking, the values for the variance (for domain truncating) that are larger than one are not significant. The results in this case are provided in Table 1, which shows the superiority of the proposed solver RBF-FD-PM.

**Example 2 ([34]).** Let us investigate the computational results of a European put option for (2) using the following settings:

$$\begin{aligned} \gamma &= -0.5, \hat{\sigma} = 0.4, \rho = -0.5, \lambda = 0.2, \sigma = 0.25, \\ r &= 0.03, T = 0.5, \kappa = 2.0, \theta = 0.04, q = 0. \end{aligned} \tag{55}$$

**Table 1.** Numerical results for Example 1.

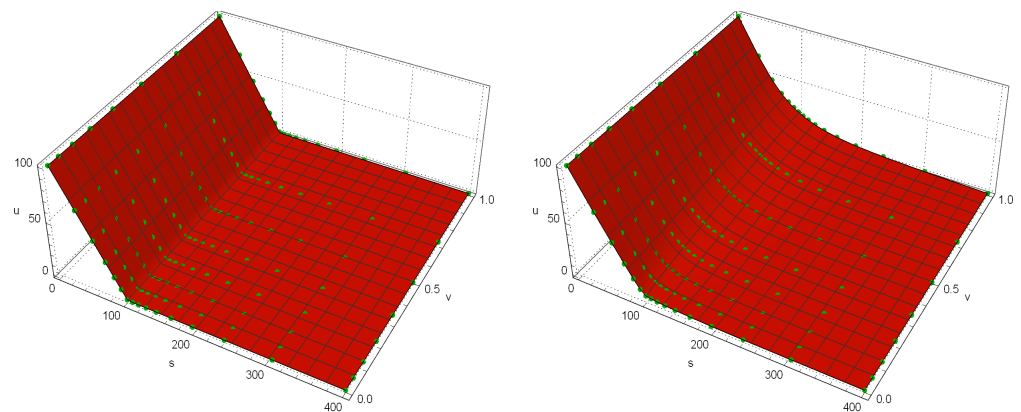
Solver	$m$	$n$	$N$	$k + 1$	$u$	$\varepsilon$	Time
FDM	20	20	400	401	8.700	1.94E-1	0.87
	40	25	1000	2001	8.597	2.97E-1	2.33
	40	40	1600	2001	8.673	2.20E-1	4.92
	65	45	2925	4001	8.860	3.39E-2	14.07
	80	50	4000	10,001	8.874	2.03E-2	31.09
AFFT	10	10	100	251	8.346	5.48E-1	0.41
	15	15	225	251	8.698	1.96E-1	0.54
	25	20	500	401	8.860	3.47E-2	0.56
	30	30	900	601	8.870	2.43E-2	1.49
	50	30	1500	2001	8.885	9.62E-3	4.46
	80	30	2400	5001	8.890	4.32E-3	11.49

**Table 1.** Cont.

Solver	$m$	$n$	$N$	$k + 1$	$u$	$\epsilon$	Time
SM	10	10	100	251	8.388	5.06E-1	0.55
	15	15	225	251	8.746	1.47E-1	0.81
	25	20	500	401	8.877	1.71E-2	1.83
	30	30	900	601	8.888	6.09E-3	3.64
	80	30	2400	2501	8.894	8.19E-4	27.19
RBF-FD-PM	10	10	100	251	8.389	5.05E-1	0.52
	15	15	225	251	5.753	1.41E-1	0.80
	25	20	500	401	8.876	1.88E-2	1.69
	30	30	900	601	8.889	5.56E-3	3.29
	80	30	2400	2501	8.894	6.69E-4	25.74

The reference prices for specific locations of the domain are 11.302917 at (90, 0.04, 0.5), 6.589881 at (100, 0.04, 0.5) and 4.191455 at (110, 0.04, 0.5) using [34]. The convergence results are provided in Tables 2 and 3 and confirm the superiority of the proposed solver with  $\psi = 2$  in this paper.

The FDM solver is back-of-the-envelope accounting because it is clear that the uniform grids for the PIDE problem are not a fast calculation to obtain highly accurate prices. To check the stability and positivity of the numerical solution for RBF-FD-PM, the numerical solution for Example 2 is plotted in Figure 1, which shows the stable behavior of RBF-FD-PM using  $m = 16$ ,  $n = 8$  and  $k = 1001$ .



**Figure 1.** Numerical solution of Example 2 using the RBF-FD-PM solver when  $\tau = 0$  on the left and  $\tau = 0.5$  on the right. Green points show the location of the graded discretization points on the red curve, which is the numerical solution.

The reason for providing Figure 1 is twofold. We must first reveal that the numerical solution obtained by RBF-FD-PM using some  $m$  and  $n$  is stable and does not have oscillations. This is important since the PIDE model has a mixed derivative term, which can lead to oscillations in the numerical solution as long as a careless numerical method is employed. Second, we must reveal how the graded meshes (the green points in Figure 1) located on the numerical solution are obtained by employing an automatic interpolation on the obtained solutions.

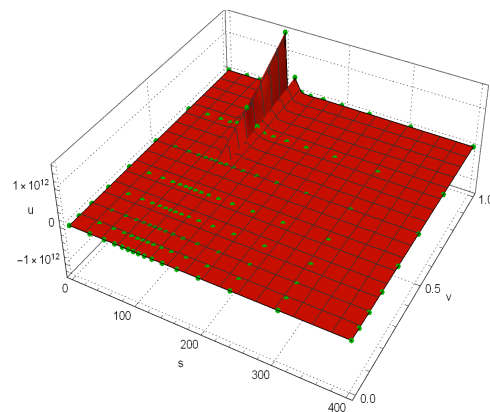
An inquiry might arise by analyzing the results in Tables 1 and 2. It is not easy to find out the advantages of the proposed approach since the numerical values are given for different values of the parameters  $m$ ,  $n$ ,  $N$  and  $k + 1$ . In fact, larger time step sizes (lower  $k$ ) are taken for SM and RBF-FD-PM since their ODE solver, i.e., (44), has a larger stability

region, and the overall solvers must be compared by fixing an accuracy for the errors and then checking the computational times.

**Table 2.** Numerical results of the different solvers in Example 2.

Solver	$m$	$n$	$N$	$k + 1$	$\epsilon$ at $x = 90$	$\epsilon$ at $x = 100$	$\epsilon$ at $x = 110$	Time
FDM								
	8	5	40	26	4.54E0	3.10E0	3.56E-1	0.37
	16	8	128	51	1.23E0	1.31E-1	1.02E0	0.52
	32	16	512	101	1.34E-1	2.77E-1	2.26E-1	1.49
	32	32	1024	201	1.25E-1	2.02E-1	2.23E-1	2.93
	64	32	2048	501	5.35E-3	5.85E-2	1.87E-2	9.47
SM								
	8	5	32	51	2.57E-1	7.48E-1	7.13E-1	0.55
	16	8	128	101	7.31E-2	2.51E-1	3.89E-2	0.81
	32	16	512	501	2.18E-3	6.02E-3	5.64E-3	2.40
	64	32	2048	1001	1.79E-3	6.24E-4	1.39E-3	16.71
RBF-FD-PM								
	8	5	32	51	2.49E-1	6.89E-1	7.11E-1	0.50
	16	8	128	101	6.25E-2	2.39E-1	3.84E-2	0.76
	32	16	512	501	2.07E-3	5.44E-3	5.04E-3	2.11
	64	32	2048	1001	1.08E-3	5.81E-4	1.04E-3	15.34

To also show how the instability may ruin the numerical pricing using the stability bound (49), we provide the numerical results of solving (2) by the RBF-FD-PM using  $m = 16$  and  $n = 8$ , but with  $k = 25$  uniform discretization nodes along time in Figure 2. This shows that all the involved solvers have some limitations, but the proposed solver sounds more efficient than others.



**Figure 2.** The unstable numerical solution of Example 2 at  $\tau = 0.5$  using  $k = 25$  nodes.

However, due to nonsmoothness at the strike price in the initial condition, it might be useful to employ a time-stepping solver that works on graded meshes over time with more focus at the beginning of the starting time, i.e., zero (the solution near the initial time point has a weak singularity). One such method is the Rannacher time-marching method [37]. Although such an application will help our solver a lot, we will try to focus on this in forthcoming related works.

**Table 3.** Numerical results of the AFFT solver in Example 2.

Solver	$m$	$n$	$\varepsilon$ at $x = 90$	$\varepsilon$ at $x = 100$	$\varepsilon$ at $x = 110$
AFFT	17	9	1.08E0	1.57E0	1.96E-1
	33	17	2.80E-2	5.20E-1	1.38E-1
	65	33	4.78E-3	1.25E-1	2.84E-2
	129	65	7.38E-3	3.09E-2	5.25E-3

## 6. Concluding Remarks

PIDEs arise in the mathematical modeling of many processes in different fields of engineering and finance. This paper has presented an approximate solution of the linear Bates PIDE with clear application in financial option pricing using a local integral term. The solution method was considered on graded meshes at which there is a clear concentration of the discretization nodes on the financially important are of the problem. Then, an RBF-FD solver using semi-discretization via sparse arrays have been constructed for solving the Bates PIDE. The numerical results were furnished and supported the theoretical discussions. These results have been provided in Tables 1 and 2 which implicitly state that the proposed approach can compete the most efficient solver (SM) for the same purpose. Additionally, the prospects for future research can be focused on how to obtain RBF-FD weights on stencils having five/six adjacent nodes on graded meshes or employing the Rannacher time-marching method in order to obtain higher accuracies for solving the PIDE problem (2).

**Author Contributions:** Conceptualization, T.L. and G.F.; formal analysis, G.F.; T.L. and S.S.; funding acquisition, M.Z.U. and T.L.; investigation, T.L., M.Z.U. and S.S.; methodology, G.F. and S.S.; supervision, T.L.; validation, M.Z.U. and S.S.; writing—original draft, G.F., M.Z.U. and S.S.; writing—review and editing, T.L., M.Z.U. and S.S. All authors have read and agreed to the published version of the manuscript.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** For the data-availability statement, we state that data sharing is not applicable to this article as no new data were created in this study.

**Acknowledgments:** The third author states that: The Deanship of Scientific Research (DSR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia, has funded this project, under grant no. (KEP-MSc: 65-130-1443). The authors are very much thankful to three anonymous referees for their suggestions, which helped to improve this paper.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- Bates, D. Jumps and stochastic volatility: The exchange rate processes implicit in Deutsche mark options. *Rev. Fin. Stud.* **1996**, *9*, 69–107. [[CrossRef](#)]
- Itkin, A. *Pricing Derivatives Under Lévy Models: Modern Finite-Difference and Pseudo-Differential Operators Approach*, Birkhäuser Basel; Springer: Berlin/Heidelberg, Germany, 2017.
- Kim, K.-H.; Sin, M.-G. Efficient hedging in general Black-Scholes model. *Finan. Math. Appl.* **2014**, *3*, 1–9.
- Ghanadian, A.; Lotfi, T. Approximate solution of nonlinear Black-Scholes equation via a fully discretized fourth-order method. *Aims Math.* **2020**, *5*, 879–893. [[CrossRef](#)]
- Heston, S. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Finan. Stud.* **1993**, *6*, 327–343. [[CrossRef](#)]
- Chang, Y.; Wang, Y.; Zhang, S. Option pricing under double Heston jump-diffusion model with approximative fractional stochastic volatility. *Mathematics* **2021**, *9*, 126. [[CrossRef](#)]
- Gómez-Valle, L.; Martínez-Rodríguez, J. Including jumps in the stochastic valuation of freight derivatives. *Mathematics* **2021**, *9*, 154. [[CrossRef](#)]
- Liu, J.; Yan, J. Convergence rate of the high-order finite difference method for option pricing in a Markov regime-switching jump-diffusion model. *Fractal Fract.* **2022**, *6*, 409. [[CrossRef](#)]

9. Chen, K.-S.; Huang, Y.-C. Detecting jump risk and jump-diffusion model for Bitcoin options pricing and hedging. *Mathematics* **2021**, *9*, 2567. [[CrossRef](#)]
10. Hellmuth, K.; Klingenberg, C. Computing Black Scholes with uncertain volatility—a machine learning approach. *Mathematics* **2022**, *10*, 489. [[CrossRef](#)]
11. Ballestra, L.; Sgarra, C. The evaluation of American options in a stochastic volatility model with jumps: An efficient finite element approach. *Comput. Math. Appl.* **2010**, *60*, 1571–1590. [[CrossRef](#)]
12. Ballestra, L.; Cecere, L. A fast numerical method to price American options under the Bates model. *Comput. Math. Appl.* **2016**, *72*, 1305–1319. [[CrossRef](#)]
13. Kluge, T. Pricing Derivatives in Stochastic Volatility Models Using the Finite Difference Method. Ph.D. Thesis, TU Chemnitz, Chemnitz, Germany, 2002.
14. in 't Hout, K.J.; Foulon, S. ADI finite difference schemes for option pricing in the Heston model with correlation. *Int. J. Numer. Anal. Model.* **2010**, *7*, 303–320.
15. Balajewicz, M.; Toivanen, J. Reduced order models for pricing European and American options under stochastic volatility and jump-diffusion models. *J. Comput. Sci.* **2017**, *20*, 198–204. [[CrossRef](#)]
16. Duffy, D. *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*; Wiley: England, UK, 2006.
17. Düring, B.; Fournié, M. High-order compact finite difference scheme for option pricing in stochastic volatility models. *J. Comput. Appl. Math.* **2012**, *236*, 4462–4473. [[CrossRef](#)]
18. Soleymani, F.; Ullah, M. A multiquadric RBF-FD scheme for simulating the financial HHW equation utilizing exponential integrator. *Calcolo* **2018**, *55*, 51. [[CrossRef](#)]
19. Milovanović, S.; von Sydow, L. A high order method for pricing of financial derivatives using radial basis function generated finite differences. *Math. Comput. Simul.* **2020**, *174*, 205–217. [[CrossRef](#)]
20. Milovanović, S.; von Sydow, L. Radial basis function generated finite differences for option pricing problems. *Comput. Math. Appl.* **2018**, *75*, 1462–1481. [[CrossRef](#)]
21. Soleymani, F.; Barfeie, M. Pricing options under stochastic volatility jump model: A stable adaptive scheme. *Appl. Numer. Math.* **2019**, *145*, 69–89. [[CrossRef](#)]
22. Soleymani, F.; Zhu, S. RBF-FD solution for a financial partial-integro differential equation utilizing the generalized multiquadric function. *Comput. Math. Appl.* **2021**, *82*, 161–178. [[CrossRef](#)]
23. Soleymani, F.; Itkin, A. Pricing foreign exchange options under stochastic volatility and interest rates using an RBF-FD method. *J. Comput. Sci.* **2019**, *37*, 101028. [[CrossRef](#)]
24. Itkin, A.; Soleymani, F. Four-factor model of quanto CDS with jumps-at-default and stochastic recovery. *J. Comput. Sci.* **2021**, *54*, 101434. [[CrossRef](#)]
25. Fasshauer, G. *Meshfree Approximation Methods with MATLAB*; World Scientific Publishing Co.: Singapore, 2007.
26. Bayona, V.; Moscoso, M.; Carretero, M.; Kindelan, M. RBF-FD formulas and convergence properties. *J. Comput. Phys.* **2010**, *229*, 8281–8295. [[CrossRef](#)]
27. Meyer, G. *The Time-Discrete Method of Lines for Options and Bonds, A PDE Approach*; World Scientific Publishing: Hackensack, NJ, USA, 2015.
28. Knapp, R. A method of lines framework in Mathematica. *J. Numer. Anal. Indust. Appl. Math.* **2008**, *3*, 43–59.
29. Salmi, S.; Toivanen, J. An iterative method for pricing American options under jump-diffusion models. *Appl. Numer. Math.* **2011**, *61*, 821–831. [[CrossRef](#)]
30. Michel, V.; Thomann, D. TVD-MOOD schemes based on implicit-explicit time integration. *Appl. Math. Comput.* **2022**, *433*, 127397.
31. Shymanskyi, V.; Protsyk, Y. Simulation of the heat conduction process in the Claydite-Block construction with taking into account the fractal structure of the material. In Proceedings of the 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 11–14 September 2018.
32. Sayfidinov, O.; Bognár, G.; Kovács, E. Solution of the 1D KPZ equation by explicit methods. *Symmetry* **2022**, *14*, 699. [[CrossRef](#)]
33. Butcher, J. *Numerical Methods for Ordinary Differential Equations*, 2nd ed.; Wiley: England, UK, 2008.
34. Salmi, S.; Toivanen, J.; von Sydow, L. An IMEX-scheme for pricing options under stochastic volatility models with jumps. *SIAM J. Sci. Comput.* **2014**, *36*, 817–834. [[CrossRef](#)]
35. Mangano, S. *Mathematica Cookbook*; O'Reilly Media: Newton, MA, USA, 2010.
36. Wellin, P.; Gaylord, R.; Kamin, S. *An Introduction to Programming with Mathematica*; Cambridge University Press: Cambridge, UK, 2005.
37. Giles, M.; Carter, R. Convergence analysis of Crank-Nicolson and Rannacher time-marching. *J. Comput. Financ.* **2006**, *9*, 89–112. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.