International Journal of Computer Mathematics
Vol. xx, No. xx, xx 201x, 1–15

RESEARCH ARTICLE

Boundary value methods with Crank-Nicolson preconditioner for option pricing model

Shu-Ling Yang[a], Spike T. Lee[b] and Hai-Wei Sun[b]*

[a]*School of Applied Mathematics, Guangdong University of Technology, China*;
[b]*Department of Mathematics, University of Macau, Macao, China*

(*Received 00 Month 200x; in final form 00 Month 200x*)

Under a jump-diffusion process, the option pricing function satisfies a partial integro-differential equation. A fourth order compact scheme is used to discretize the spatial variable of this equation. The boundary value method is then applied for temporal direction because of its unconditional stability. To avert the numerical oscillation caused by the non-smooth payoff, both the second order backward difference formula and the boundary value method are adopted on the initial time layer. Moreover, the resulting linear system of the boundary value method applied in the follow-up layers is solved by the GMRES method with a preconditioner which comes from the Crank-Nicolson scheme. With regard to this preconditioner, studies for invertibility and convergence in right-preconditioned GMRES method are given. Numerical experiments demonstrate the superiority of the method.

Keywords: jump-diffusion; boundary value method; second order backward difference formula; preconditioner; Crank-Nicolson scheme

AMS Subject Classification: 65M06, 65L05, 65F10, 15A09, 91B28

1. Introduction

The reason why we study jump-diffusion models [9, 12] is that they depict the price in real market more accurately than the Black-Scholes model [3], especially when the stock price changes sharply. In Merton's model, the changing stock price was assumed to follow a standard Wiener process with a jump-diffusion component. Jumps occur according to a compound Poisson process with constant intensity and normally distributed jump sizes. For European call and put options, there exists an analytical solution in Merton's model. However, it is not the same case for some complicated options like path-dependent options. Therefore, using a numerical approach to evaluate the option prices is very necessary. The price of an option $u(x,t)$ under the jump-diffusion process satisfies a partial integro-differential equation (PIDE) [1, 6, 7, 11]:

$$u_t = \frac{\sigma^2}{2}u_{xx} + (r - \lambda\kappa - \frac{\sigma^2}{2})u_x - (r+\lambda)u + \lambda\int_{-\infty}^{\infty} u(x+z,t)\phi(z)\mathrm{d}z, \qquad (1)$$

where $x \in (-\infty, +\infty) = \mathbb{R}$ is the logarithmic price, $t \in [0,T]$ is the time to maturity T. The option pricing parameters are as follows: $\sigma > 0$ is the stock return volatility,

* Corresponding author. Email: HSun@umac.com

ISSN: 0020-7160 print/ISSN 1029-0265 online

DOI: 0020716YYxxxxxxxx
http://www.informaworld.com

$r \geq 0$ is the risk-free interest rate, λ is the arrival intensity of a Poisson process, κ is the expectation of the impulse function, and $\phi(z)$ is the density function.

There are numerous ways to solve a PIDE like (1) [1, 6, 7]. However, most existing methods can only reach second order accuracy in space because of the non-smooth property of payoff. Tangman [15] obtained a fourth order accurate solution of PDE by a high order compact finite difference scheme with grid stretching technique, but it is only applied to the Black-Scholes model. Recently, Lee and Sun [11] proposed to use the fourth order compact (FOC) scheme in space and boundary value method (BVM) in time. The non-smoothness is tackled by local mesh refinement in space and second order backward difference formula (BDF2) startup procedure in time. Although this approach, called FOCBVM, ultimately reaches fourth order convergence order in both spatial and temporal direction, it is still bothered by the following issues. Because of the increasing node numbers in time direction, the operation cost of BDF2 startup with small step size is comparatively huge. Besides, the iteration number of GMRES method with a circulant-type preconditioner applied to solve the resulting linear system gradually increases too as the grid is refined.

In this paper, we consider a rapid high order accurate method for solving (1). The detailed procedure is as follows. We first discretize (1) by FOC scheme with local mesh refinement based on a three point stencil in spatial direction. As a result of discretization, an initial value problem (IVP) of ODEs is obtained. We then get the approximation on the initial time layer by applying two methods with small step size: the BDF2 and the BVM. This treatment can efficiently avoid numerical disasters brought by the non-smooth property of initial value. Then the same BVM is also implemented on the following time layers. We solve the linear system in the BVM by GMRES method with a preconditioner different from [11]. This preconditioner comes from the coefficient matrix of Crank-Nicolson discretization scheme [8]. Numerical results show that our approach is easy to realize and gets high order accuracy for option pricing models.

The remainder of this paper is organized as follows. In Section 2, we discretize the corresponding PIDE in space by FOC difference scheme with local mesh refinement. In Section 3 we introduce the BDF2 and BVMs for the semi-discretized system. A preconditioner is constructed for the resulting system in Section 4. We prove the invertibility of this preconditioner and estimate the convergence of the first iteration in GMRES method. Numerical results are presented in Section 5. In Section 6, we give a conclusion and some ideas for future work.

2. FOC scheme for PIDE

2.1 *FOC scheme*

We consider the jump-diffusion model proposed by Merton [12], where jumps are normally distributed with mean μ and variation γ, and the density function ϕ is the Gaussian distribution:

$$\phi(z) = \frac{e^{-(z-\mu)^2/2\gamma^2}}{\sqrt{2\pi}\gamma}. \tag{2}$$

For European call options, which we are interested in, the initial condition of equation (1) is the payoff function:

$$u(x, 0) = \max\{K(e^x - 1), 0\}. \tag{3}$$

Boundary conditions of equation (1) are given by:

$$u(x,t) \approx \begin{cases} 0, & x \to -\infty, \\ K(e^x - e^{-rt}), & x \to +\infty. \end{cases} \tag{4}$$

We need to solve the PIDE (1) with initial condition (3) and boundary conditions (4) to obtain the option value $u(x,T)$ [1, 6, 7].

For European call options in jump-diffusion models, the state space $\mathbb{R}$ is unbounded. In order to develop the numerical calculation of models with boundary conditions (4), we also need to employ the local truncation method [7]. Let the unbounded domain $\mathbb{R}$ be truncated to a bounded domain $[x_{\min}, x_{\max}]$. We then divide this domain with a constant space step:

$$h_x = \frac{x_{\max} - x_{\min}}{m+1}, \ x_i = x_{\min} + i \cdot h_x, \ i = 0, \ldots, m+1.$$

Notice that the number of inner grid nodes is m, and we assume that the mesh grid includes the point $x^* = 0$, which makes $u(x,0)$ non-smooth around x^* in the initial condition.

We then move on to demonstrate the spatial discretization of (1) by FOC scheme, the readers are referred to see [11] for details. We first denote the coefficients in (1) as

$$a_1 = \frac{1}{2}\sigma^2 > 0, \quad a_2 = r - \lambda\kappa - \frac{\sigma^2}{2} \quad \text{and} \quad a_3 = r + \lambda \geq 0.$$

Applying the FOC scheme to (1), we obtain two coefficient matrices related to the FOC discretization [11]:

$$L_x = \frac{1}{12}\text{tridiag}\left[1 - \frac{a_2 h_x}{2a_1}, 10, 1 + \frac{a_2 h_x}{2a_1}\right] \tag{5}$$

and

$$\begin{aligned} R = \text{tridiag}\Bigg[& \frac{a_2^2}{12a_1} + \frac{a_1}{h_x^2} - \frac{a_3}{12} - \left(\frac{a_2}{2h_x} - \frac{a_2 a_3 h_x}{24a_1}\right), -\frac{a_2^2}{6a_1} - \frac{2a_1}{h_x^2} - \frac{5a_3}{6}, \\ & \frac{a_2^2}{12a_1} + \frac{a_1}{h_x^2} - \frac{a_3}{12} + \left(\frac{a_2}{2h_x} - \frac{a_2 a_3 h_x}{24a_1}\right)\Bigg]. \end{aligned} \tag{6}$$

We remark that L_x and R are $m \times m$ tridiagonal matrices. In particular, the diagonal entries of L_x are always positive, while those of R are always negative.

For the integral term in (1), we split it as [1, 11]:

$$\begin{aligned} \int_{-\infty}^{\infty} u(x+z,t)\phi(z)dz = & \int_{-\infty}^{\infty} u(z,t)\phi(z-x)dz \\ = \int_{x_1}^{x_m} u(z,t)\phi(z-x)dz & + \int_{[x_1,x_m]^c} u(z,t)\phi(z-x)dz, \end{aligned}$$

where $[x_1, x_m]^c = \mathbb{R} \backslash [x_1, x_m]$. For calculation of the first integral on $[x_1, x_m]$, we consider the fourth order accurate composite Simpson's rule. The relevant coeffi-

cient matrix is an $m \times m$ Toeplitz-like matrix denoted by

$$\Phi = \frac{h_x}{3} \left[\phi(x_k - x_i)\right]_{i,k} \cdot \mathrm{diag}[1, 4, 2, 4, 2, \ldots, 1].$$

On the other hand, $u(z,t)$ can be approximated by boundary conditions (4) on $[x_1, x_m]^{\mathrm{c}}$, hence the second integral term for each $x = x_i,\ i = 1, 2, \ldots, m$ is approximately rewritten as [1, 11]:

$$\xi_i(t) \equiv Ke^{x_i+\mu+\gamma^2/2}\mathcal{N}\left(\frac{x_i - x_m + \mu}{\gamma} + \gamma\right) - Ke^{-rt}\mathcal{N}\left(\frac{x_i - x_m + \mu}{\gamma}\right),$$

where $\mathcal{N}$ is the standard normal cumulative distribution function.

Let $u_i(t)$ be the discrete approximate value of $u(x_i, t)$. Define the vector mark

$$\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_m(t)]^{\intercal}.$$

After all, the semi-discretized ODE system is

$$[L_x\mathbf{u}(t) + \mathbf{c}_1(t)]' = R_x\mathbf{u}(t) + \mathbf{c}_2(t), \tag{7}$$

where R_x is the sum of R, $\lambda L_x \cdot \Phi$ and a rank-1 matrix, $\mathbf{c}_1(t)$ and $\mathbf{c}_2(t)$ are vectors depending on variable t, and $\mathbf{c}_2(t)$ mainly consists of the values $\xi_i(t)$. See [11] for details.

2.2 *Local mesh refinement*

Because of the non-smooth characteristic of payoff function for European call options, the FOC scheme on a uniform grid does not achieve fourth order accuracy in numerical experiments [10]. A grid stretching transformation is often used to restore accuracy by concentrating grid nodes at the non-smooth point $x^* = 0$. However, this method can not keep any good quality of the coefficient matrices in spatial discretization scheme. In this paper, we also apply the local mesh refinement technique [10, 11, 18]. This procedure has the advantage of easy collaboration with the three point stencil of FOC scheme, and keeps the features of the coefficient matrices.

In the beginning, the bounded domain $[x_{\min}, x_{\max}]$ is divided into $m + 1$ subdomain with a fixed step size h_x. Then we apply the local mesh refinement around the non-smooth point $x^* = 0$. After this treatment, the spatial mesh includes $l = m + 4\lceil -\log_2 h_x \rceil$ inner nodes, where the operator $\lceil x \rceil$ rounds x to the nearest integer toward the positive infinity [10, 11].

We employ the FOC scheme to the equation (1) on the new refined mesh again. After applying the FOC scheme, we denote $\hat{L}_x, \hat{R} \in \mathbb{R}^{l\times l}$ as the coefficient matrices of the FOC discretization, which in fact are the refined version of L_x and R. It is noted that $\hat{L}_x$ and $\hat{R}$ are tridiagonal-like matrices. The splitting method is then adopted to calculate the integral term as before. After coping with the spatial discretization and the integral term, we transform the PIDE (1) with initial condition (3) and boundary conditions (4) to an IVP of ODEs:

$$\begin{cases} [\hat{L}_x\hat{\mathbf{u}}(t) + \hat{\mathbf{c}}_1(t)]' = \hat{R}_x\hat{\mathbf{u}}(t) + \hat{\mathbf{c}}_2(t), & t \in [0, T], \\ \hat{\mathbf{u}}(0) = \hat{\mathbf{d}}, \end{cases} \tag{8}$$

where the definitions of all $\hat{R}_x$, $\hat{\mathbf{c}}_1(t)$, $\hat{\mathbf{c}}_2(t)$ and $\hat{\mathbf{u}}(t)$ are analogous to those in equation (7), and $\hat{\mathbf{d}} \in \mathbb{R}^l$ is the vector of initial values on spatial grid nodes. See the details in [11].

3. BVMs with BDF2 and BVM startup

In this section, we consider how to solve (8) by BVMs, which are numerical methods based on the linear multistep formula (LMF) and renowned for high order accuracy and unconditional stability [2, 4]. Corresponding to the fourth order accurate method in spatial direction, the BVM is applied to discretize ODEs in time direction.

3.1 *FOCBVM for ODE system*

We first consider solving the following general IVP:

$$\begin{cases} y'(t) = f(t, y), & t \in [0, T], \\ y(0) = y_0. \end{cases} \tag{9}$$

by BVMs. We use a uniform mesh over the time interval. The step size is assumed to be $h_t = T/n$, then the nodes are: $t_j = j \cdot h_t,\ j = 0, 1, \ldots, n$. Let $y_j,\ f_j$ be the approximation to $y(t_j)$ and $f(t_j, y(t_j))$ respectively. A p-step BVM requires the following equations which come from a p-step LMF:

$$\sum_{i=-q}^{p-q} \alpha_{i+q} y_{j+i} = h_t \sum_{i=-q}^{p-q} \beta_{i+q} f_{j+i}, \quad j = q, \ldots, n-p+q. \tag{10}$$

Since only the initial value y_0 is given, the equations in (10) should be coupled with q initial conditions and $(p-q)$ final conditions. The following $(p-1)$ equations are used to obtain the other initial and final values:

$$\sum_{i=0}^{p} \alpha_i^{(k)} y_i = h_t \sum_{i=0}^{p} \beta_i^{(k)} f_i, \quad k = 1, \ldots, q-1, \tag{11}$$

and

$$\sum_{i=0}^{p} \alpha_{p-i}^{(k)} y_{n-i} = h_t \sum_{i=0}^{p} \beta_{p-i}^{(k)} f_{n-i}, \quad k = n-p+q+1, \ldots, n. \tag{12}$$

Note that the coefficients $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ in (11) and (12) are determined by difference schemes with truncation errors consistent with (10). Applying these formulas to (9), we obtain a linear system in matrix form:

$$L_p \mathbf{y} = h_t R_p \mathbf{f} + \mathbf{e}_1 y_0 + \mathcal{O}(h_t^{p+1}),$$

where $\mathbf{y} = [y_0, y_1, \ldots, y_n]^\intercal$, $\mathbf{f} = [f_0, f_1, \ldots, f_n]^\intercal$, $\mathbf{e}_1 = [1, 0, \ldots, 0]^\intercal \in \mathbb{R}^{(n+1)}$, and L_p, R_p are $(n+1) \times (n+1)$ matrices which elements depend on the applied BVM.

For instance, the matrix L_p has the following form:

$$L_p = \begin{bmatrix} 1 & \cdots & 0 & & & & \\ \alpha_0^{(1)} & \cdots & \alpha_p^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & & \\ \alpha_0^{(q-1)} & \cdots & \alpha_p^{(q-1)} & & & & \\ \alpha_0 & \cdots & \alpha_p & & & & \\ & \alpha_0 & \cdots & \alpha_p & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \alpha_0 & \cdots & \alpha_p & \\ & & & \alpha_0^{(n-p+q+1)} & \cdots & \alpha_p^{(n-p+q+1)} & \\ & & & \vdots & \vdots & \vdots & \\ & & & \alpha_0^{(n)} & \cdots & \alpha_p^{(n)} & \end{bmatrix},$$

while the matrix R_p is similar to L_p but with all α's replaced by β's, and the entries in the first row are all zeros. See [2, 4, 5] for details.

Now we return to our FOC-discretized system (8), which is an IVP of ODEs. Setting $\overline{\mathbf{u}}_j$ as the discrete approximation of $\hat{\mathbf{u}}(t_j)$, we denote

$$\begin{aligned} \overline{\mathbf{u}} &= [\overline{\mathbf{u}}_0^\intercal, \overline{\mathbf{u}}_1^\intercal, \ldots, \overline{\mathbf{u}}_n^\intercal]^\intercal, \\ \overline{\mathbf{c}}_1 &= [\hat{\mathbf{c}}_1^\intercal(t_0), \hat{\mathbf{c}}_1^\intercal(t_1), \ldots, \hat{\mathbf{c}}_1^\intercal(t_n)]^\intercal, \\ \overline{\mathbf{c}}_2 &= [\hat{\mathbf{c}}_2^\intercal(t_0), \hat{\mathbf{c}}_2^\intercal(t_1), \ldots, \hat{\mathbf{c}}_2^\intercal(t_n)]^\intercal \end{aligned}$$

and $\zeta = \hat{L}_x \cdot \hat{\mathbf{d}} + \hat{\mathbf{c}}_1(t_0)$. Applying a p-step BVM to (8), we arrive at the following linear system:

$$\begin{aligned} A_p\overline{\mathbf{u}} &\equiv (L_p \otimes \hat{L}_x - h_t \cdot R_p \otimes \hat{R}_x)\overline{\mathbf{u}} \\ &= h_t \cdot (R_p \otimes I_l)\overline{\mathbf{c}}_2 + \mathbf{e}_1 \otimes \zeta - (L_p \otimes I_l)\overline{\mathbf{c}}_1 \equiv b, \end{aligned} \tag{13}$$

where the suffix "p" in A_p implies that a p-step BVM is used, $\otimes$ is the Kronecker product, and I_l is an $l \times l$ identity matrix. See the details in [2, 5, 11].

3.2 *BDF2 and BVM startup*

It was shown that the unconditionally stable Crank-Nicolson scheme has oscillatory results when it is applied to compute the option prices in the Black-Scholes model [16]. The reason is the non-smoothness of payoff function. Furthermore, the Crank-Nicolson scheme does not converge with second order accuracy as the time mesh size decreases. The similar bad result happens to BVMs when numerically solving the ODE system (8) in this paper.

Rannacher [13] proposed a method to restore the second order convergence of Crank-Nicolson scheme. He replaced the first time layer by two time layers with half step size. Lee and Sun used a similar method in [11], and the idea is as follows. Let the initial time step size be h_t. The BDF2 is applied to compute the approximate price values in the first two time layers $[0, 2h_t]$ with small time step size $\hat{h}_t \leq h_t^2$. For sequent time layers $[2h_t, T]$, a p-step BVM is then adopted. This procedure efficiently avoids the oscillation problem and restores the accuracy of p-step BVM because the small step size $\hat{h}_t$ is mostly adopted in the first two time layers.

However, the small step size $\hat{h}_t$ becomes very small when the time mesh size h_t decreases. Thus the operation cost of BDF2 startup method applied on $[0, 2h_t]$ becomes very large. To be specific, dividing $[0, 2h_t]$ with a step size $\hat{h}_t \leq h_t^2$ results in

$$\frac{2h_t}{\hat{h}_t} \geq \frac{2h_t}{h_t^2} = \frac{2}{h_t} = \frac{2n}{T} = \mathcal{O}(n)$$

time layers, i.e., we have $\mathcal{O}(n)$ systems to solve before applying the p-step BVM on $[2h_t, T]$. In particular, for short-maturity options with $T < 1$, the situation is even worsened since the number of time layers in $[0, 2h_t]$ gets amplified by T. In this paper, we improve the BDF2 startup by the BDF2 and BVM startup on $[0, h_t]$ to overcome the shortcomings of the method in [11].

For brevity, we illustrate the BDF2 and BVM startup by solving the following ODE:

$$\begin{cases} y'(t) = f(t, y), & t \in [0, h_t], \\ y(0) = y_0. \end{cases} \tag{14}$$

To start with, we choose a small step size $\hat{h}_t$. The choice of this small step size $\hat{h}_t$ must be consistent with the convergence accuracy of BVM applied on $[h_t, T]$. For example, we define $\hat{h}_t \leq h_t^2$ for a fourth order accurate BVM. Set a parameter $s = \lceil \log(1/h_t) \rceil$, where the operator $\lceil \cdot \rceil$ is defined as before. The BDF2 is first applied on the interval $[0, (c \cdot s)\hat{h}_t]$ divided by the fixed step size $\hat{h}_t$:

$$0 = \hat{t}_0 < \hat{t}_1 < \hat{t}_2 < \hat{t}_3 < \cdots < \hat{t}_{c \cdot s} = (c \cdot s)\hat{h}_t.$$

It is noticed that c is a positive integer constant and is chosen in accordance with

$$c \cdot \log(1/h_t) \ll 1/h_t.$$

We apply the implicit Euler scheme to calculate the value y_1, which is the approximate solution to $y(\hat{t}_1)$ of ODE (14) at $\hat{t}_1 = \hat{h}_t$. Then we compute the approximate value y_j of $y(\hat{t}_j)$ using the following BDF2 formula:

$$\frac{3\, y_j \; - \; 4\, y_{j-1} \; + \; y_{j-2}}{2\hat{h}_t} = f(t_j, y_j), \quad j = 2, 3, \cdots, c \cdot s.$$

For the remaining domain $[(c \cdot s)\hat{h}_t, h_t]$, we divide it equally with $s - 1$ nodes and utilize the BVM in §3.1 on this time mesh. Note that the BDF2 solution at time $(c \cdot s)\hat{h}_t$ becomes the new initial value of BVM on $[(c \cdot s)\hat{h}_t, h_t]$. Ultimately, we obtain the final approximation at time h_t for ODE (14).

For our original semi-discretized problem (8), according to the above procedure, we first solve (8) on $[0, (c \cdot s)\hat{h}_t]$ instead of $[0, T]$, and eventually obtain the approximation to $\hat{\mathbf{u}}((c \cdot s)\hat{h}_t)$. Secondly, we employ the BVM on time domain $[(c \cdot s)\hat{h}_t, h_t]$, which is equally divided by $s - 1$ nodes. After the above BDF2 and BVM startup, the approximate value $\overline{\mathbf{u}}_1$ of $\hat{\mathbf{u}}(h_t)$ is computed. The same BVM method is then applied to solve (8) on time interval $[h_t, T]$. Note that the initial value of (8) is replaced by $\overline{\mathbf{u}}_1$ and the mesh nodes associated with the BVM have decreased by one.

In [11], the number of time mesh nodes in $[0, 2h_t]$ is of $\mathcal{O}(n)$, where n is the number of layers in $[0, T]$. On the other hand, the corresponding node number in

the BDF2 and BVM startup method proposed by us is of

$$c \cdot s + s = (c+1) \cdot s = (c+1) \cdot \lceil \log(1/h_t) \rceil = \mathcal{O}(\log n).$$

The new method hence largely decreases the operation cost of startup procedure. Moreover, the new method effectively evades the oscillation result and recovers convergence accuracy of BVMs, just like the original BDF2 startup method. We later refer to the numerical experiments in Section 5.

4. Preconditioner from Crank-Nicolson scheme

4.1 *The form of Crank-Nicolson preconditioner*

Note that the coefficient matrix A_p in linear system (13) is very large when step sizes h_x and h_t are refined. Therefore it is difficult and expensive to solve this linear system directly. Considering the sparsity of matrix A_p, we apply the iterative method with preconditioning technique to solve the linear system. A large number of preconditioners such as circulant-like preconditioners are often used [2, 5]. The operation cost of inverting an $N \times N$ circulant-like preconditioner is $\mathcal{O}(N \log N)$. In this section, we propose to apply a preconditioner which comes from the matrix form of Crank-Nicolson scheme [8].

Consider a constant coefficient linear ODE system:

$$\begin{cases} \mathbf{v}'(t) = J \cdot \mathbf{v}(t) + \mathbf{g}(t), & t \in [0, T], \\ \mathbf{v}(0) = \mathbf{v}_0. \end{cases} \tag{15}$$

where $\mathbf{v}(t),\ \mathbf{g}(t) : \mathbb{R} \to \mathbb{R}^m$, $\mathbf{v}_0 \in \mathbb{R}^m$ and $J \in \mathbb{R}^{m \times m}$ is the constant coefficient matrix.

Suppose the time interval $[0, T]$ is divided by a fixed step size $h_t = T/n,\ t_j = j \cdot h_t,\ j = 0, 1, \ldots, n$. Let $\mathbf{v}_j$ denote the approximation of $\mathbf{v}(t_j)$, we have the following Crank-Nicolson scheme for the ODE system (15):

$$\mathbf{v}_{j+1} - \mathbf{v}_j = \frac{h_t}{2}[J \cdot \mathbf{v}_{j+1} + \mathbf{g}(t_{j+1}) + J \cdot \mathbf{v}_j + \mathbf{g}(t_j)].$$

We change the above scheme into the matrix form, then obtain the coefficient matrix with I_m being an $m \times m$ identity matrix:

$$C = C_L \otimes I_m - h_t \cdot C_R \otimes J, \tag{16}$$

where $C_L, C_R \in \mathbb{R}^{(n+1)\times(n+1)}$ are bidiagonal matrices:

$$C_L = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, C_R = \begin{bmatrix} 0 & 0 & & \\ 1/2 & 1/2 & & \\ & \ddots & \ddots & \\ & & 1/2 & 1/2 \end{bmatrix}.$$

As we all know, the Crank-Nicolson scheme has an advantage of being unconditionally stable with second order accuracy. It can be treated as a 1-step BVM and is easily solved step by step [4]. Hence, in order to solve the linear system (13) discretized by the BVMs, we propose to use the following preconditioner, which is

inspired by the form of coefficient matrix C in Crank-Nicolson scheme [8]:

$$\hat{P}_c = C_L \otimes \hat{L}_x - h_t \cdot C_R \otimes \hat{R}. \tag{17}$$

Note that C_L, C_R are now $\mathbb{R}^{n\times n}$ matrices owing to the BDF2 startup. Also we choose the matrix $\hat{R}$ instead of $\hat{R}_x$, where $\hat{R}_x$ is the sum of $\hat{R}$ and some other matrices produced by the calculation of integral term. It is because we want to get rid of the integral part for convenience when we construct the preconditioner.

4.2 *The invertibility of Crank-Nicolson preconditioner*

We give the invertibility theory of Crank-Nicolson preconditioner in this subsection. For simplicity of discussion, here we only consider the case of uniform spatial mesh. In this case, the preconditioner is written as:

$$P_c = C_L \otimes L_x - h_t \cdot C_R \otimes R. \tag{18}$$

We have the following invertibility theory for P_c in (18).

THEOREM 4.1 *The matrix L_x in* (5) *is a nonsingular matrix, therefore the inverse L_x^{-1} exists. Furthermore, let $\lambda_j(R\cdot L_x^{-1})$, $j = 1, \ldots, m$ be the eigenvalues of $R\cdot L_x^{-1}$. For any $j = 1, \ldots, m$, $\lambda_j(R \cdot L_x^{-1})$ is not a non-negative real number.*

Proof From (5) we know that $(L_x + L_x^\intercal)/2$ is diagonally dominant and has positive diagonal entries, hence the eigenvalues of $(L_x+L_x^\intercal)/2$ are positive. Since $(L_x+L_x^\intercal)/2$ is the symmetric part of L_x, we derive that L_x is a nonsingular matrix.

For convenience, we consider the eigenvalues of $L_x^{-1} \cdot R$, which shares the same eigenvalues with $R \cdot L_x^{-1}$. Suppose $L_x^{-1} \cdot R \cdot \eta = \lambda_0\eta$, where λ_0 is a non-negative real number and η is a vector not equal to zero. We therefore have

$$\eta^\intercal(R + R^\intercal)\eta = \lambda_0\eta^\intercal(L_x + L_x^\intercal)\eta, \tag{19}$$

From (6), we recall that the matrix R is the tridiagonal coefficient matrix of FOC scheme, its diagonal entries are always negative. Therefore the diagonal entries of $R+R^\intercal$ are also negative, and it is easy to show that $R+R^\intercal$ is diagonally dominant. Then we have that $R + R^\intercal$ is negative definite:

$$\eta^\intercal(R + R^\intercal)\eta < 0. \tag{20}$$

Moreover, in the beginning of the proof we already have that $L_x + L_x^\intercal$ is positive definite:

$$\eta^\intercal(L_x + L_x^\intercal)\eta > 0. \tag{21}$$

Substituting (20) and (21) into (19) gives the contradiction. Hence $\lambda_j(R \cdot L_x^{-1})$ is not a non-negative real number for any $j = 1, \ldots, m$. ■

We now show that the Crank-Nicolson preconditioner P_c (18) is nonsingular. Using some properties of Kronecker product, we simplify P_c as

$$P_c = [C_L \otimes I_m - h_t \cdot C_R \otimes (R \cdot L_x^{-1})] \cdot (I_n \otimes L_x) \equiv P_c^* \cdot (I_n \otimes L_x).$$

Therefore, we only need to study the invertibility of the matrix P_c^*. Note that it is easy to get the eigenvalues of C_L, C_R from their explicit forms [8], hence the

eigenvalues of P_c^* are

$$\lambda_{ij}(P_c^*) = \begin{cases} 1, & i = 0,\ j = 1, \dots m, \\ 1 - \dfrac{h_t}{2} \cdot \lambda_j(R \cdot L_x^{-1}), & i = 1, \dots, n-1,\ j = 1, \dots, m. \end{cases}$$

According to Theorem 4.1, we have $h_t \cdot \lambda_j(R \cdot L_x^{-1}) \neq 2$ for any $j = 1, \dots, m$. Thus P_c^* is nonsingular, which implies that P_c is also nonsingular. See [8] for details.

4.3 *The convergence of Crank-Nicolson preconditioner*

We now consider how to solve the linear system (13) based on p-step BVMs. Here we opt for the right-preconditioned GMRES method, where the 2-norm of residuals in each iteration is not impacted by the preconditioning operation [14]. Let v be the restriction of the continuous solution of (8) on the time mesh. According to the definition of order of p-step BVM [4], we have

$$r_p = A_p v - b \quad \text{and} \quad ||r_p||_\infty = \mathcal{O}(h_t^{p+1}). \tag{22}$$

Moreover, for a different $\hat{p}$-step BVM ($\hat{p} < p$), Brugnano and Trigiante [4] proved the following result:

$$||A_p A_{\hat{p}}^{-1} r_{\hat{p}} - r_{\hat{p}}||_\infty = \mathcal{O}(h_t^{\hat{p}+2}). \tag{23}$$

We then have the following theorem for the preconditioner $\hat{P}_c$.

THEOREM 4.2 *Let $\hat{P}_c$ given by* (17) *be the right preconditioner for solving* (13) *by the GMRES method. For any $p \geq 2$, we have*

$$||A_p \hat{P}_c^{-1} b - b||_\infty = \mathcal{O}(h_t^3).$$

Proof Note that the Crank-Nicolson discretization scheme is in fact a 1-step BVM with order 3, hence we have the follow result by the definition of order:

$$r_c = \hat{P}_c v - b \quad \text{and} \quad ||r_c||_\infty = \mathcal{O}(h_t^3). \tag{24}$$

Putting (22), (23) and (24) together gives

$$\begin{aligned} ||A_p \hat{P}_c^{-1} b - b||_\infty &= ||A_p v - A_p \hat{P}_c^{-1} r_c - b||_\infty \\ &= ||r_p - r_c - (A_p \hat{P}_c^{-1} r_c - r_c)||_\infty = \mathcal{O}(h_t^3) \end{aligned}$$

■

In order to show that the Crank-Nicolson preconditioner is well selected, we proceed to study the convergence of the right-preconditioned GMRES method. In fact, with the presence of Theorem 4.2, we can directly deduce the residual in the first iteration. Recall the right-preconditioned GMRES algorithm, the first iteration starts with letting v_0 be the initial iterative vector. Then the following

terms are computed one by one:

$$\begin{cases} \beta = b - A_p v_0, \\ v_1 = \beta/||\beta||_2, \\ \omega = A_p \hat{P}_c^{-1} v_1, \\ \omega_1 = \omega - (\omega, v_1) v_1. \end{cases}$$

We give the following theorem to estimate the residual of the first iteration $||\omega_1||_2/||\omega||_2$.

THEOREM 4.3 *The residual of the first iteration $||\omega_1||_2/||\omega||_2$ is about $\mathcal{O}(\sqrt{l/n^5})$, where l is the number of grid points on the refined spatial mesh, and n is the number of time layers.*

Proof According to the algorithm of right-preconditioned GMRES method, we know that $(\omega_1, v_1) = 0$, thus

$$\begin{aligned} ||\omega_1||_2 &\le ||\omega_1 + (\omega, v_1)v_1 - v_1||_2 = ||\omega - (\omega, v_1)v_1 + (\omega, v_1)v_1 - v_1||_2 \\ &= ||\omega - v_1||_2 = ||A_p \hat{P}_c^{-1} v_1 - v_1||_2. \end{aligned}$$

By Theorem 4.2, it follows that

$$||\omega_1||_2 \le \mathcal{O}(\sqrt{ln}\, h_t^3) = \mathcal{O}(\sqrt{l/n^5}).$$

In addition,

$$\begin{aligned} ||\omega||_2 = ||A_p \hat{P}_c^{-1} v_1||_2 &\ge ||v_1||_2 - ||A_p \hat{P}_c^{-1} v_1 - v_1||_2 \\ &= 1 - \mathcal{O}(\sqrt{ln}\, h_t^3) = 1 - \mathcal{O}(\sqrt{l/n^5}). \end{aligned}$$

Hence we get

$$\frac{||\omega_1||_2}{||\omega||_2} \le \frac{\mathcal{O}(\sqrt{l/n^5})}{1 - \mathcal{O}(\sqrt{l/n^5})} \approx \mathcal{O}(\sqrt{l/n^5}), \quad n \to +\infty.$$

■

Based on the properties of Crank-Nicolson preconditioner, we have proved that the GMRES method with Crank-Nicolson preconditioner will converge very fast as $n \to \infty$ with l fixed. We remark that only the behavior of the first iteration is studied in this paper. In fact, the GMRES method with Crank-Nicolson preconditioner performs consistently well after the first iteration, thus results in a smaller iteration number. We will illustrate this phenomenon by numerical experiments in the next section.

5. Numerical results

In this section, we first verify the effect of BDF2 and BVM startup with small step size by numerical experiments. Iterative numbers of different preconditioners in the preconditioned GMRES method are also compared. For simplicity, we consider the pricing problem of European call options in Merton's jump-diffusion model, where an analytic solution is provided in [12]. The relevant parameters are selected

as: $r = 0.05$, $\sigma = 0.25$, $\lambda = 0.1$, $T = 0.25$, $K = 100$, $\mu = -0.9$, $\gamma = 0.45$, $\kappa = e^{(\mu+\gamma^2/2)} - 1$, and $c = 6$.

The fourth order ETR_2 method, which is a kind of BVMs [4], is utilized to match up the FOC difference scheme in spatial direction. We use the right-preconditioned GMRES method to solve the linear system (13) in the following experiments. The initial guess vector for the iterative solver is zero. The stopping criterion is $||\mathbf{r}_k||_2/||\mathbf{r}_0||_2 < 5\text{e}^{-13}$, where $\mathbf{r}_k$ is the true residual after k iterations.

Recall that m is the number of inner grid nodes before local mesh refinement, and n is the number of time layers. We denote ε_{h_x,h_t} as the maximum absolute error between the approximation with step size h_x, h_t and the true solution at time T. The "order" is defined as the $\log_2$-ratio of ε_{h_x,h_t} to $\varepsilon_{h_x/2,h_t/2}$, and n^* is the number of time layers in $[0, h_t]$ on which the startup procedure is applied.

We first investigate how the BDF2 and BVM startup differs from the one in [11] by measuring the CPU time spent to obtain the option price at time T. The resulting BVM systems are solved by the right-preconditioned GMRES method with Crank-Nicolson preconditioner. Table 1 shows that the number of time layers n^* in the BDF2 and BVM startup does not rapidly increase when the step size decreases in time direction. Only $n^* = \mathcal{O}(\log n)$ time layers are actually needed. On the other hand, the corresponding number related to the old BDF2 startup is of $\mathcal{O}(n)$, as we have previously analyzed. We can see from both Table 1 and Figure 1 that the BDF2 and BVM startup remarkably lowers the operation cost and saves a lot of CPU time on the premise that the convergence order is kept unchanged.

Table 1. Errors and convergence orders of FOCBVM respectively with BDF2 startup and BDF2+BVM startup for pricing European call options in Merton's model.

m	n	BDF2 n^*	startup CPU	BDF2+BVM startup n^*	CPU	ε_{h_x,h_t}	order
21	5	40	0.055	21	0.029	2.19e-2	-
45	10	80	0.121	28	0.041	4.66e-3	2.23
101	20	160	0.239	35	0.194	3.86e-4	3.59
233	40	320	1.185	42	0.415	2.77e-5	3.80
481	80	640	9.785	42	1.020	1.88e-6	3.88
1089	160	1280	74.643	49	4.136	1.22e-7	3.94

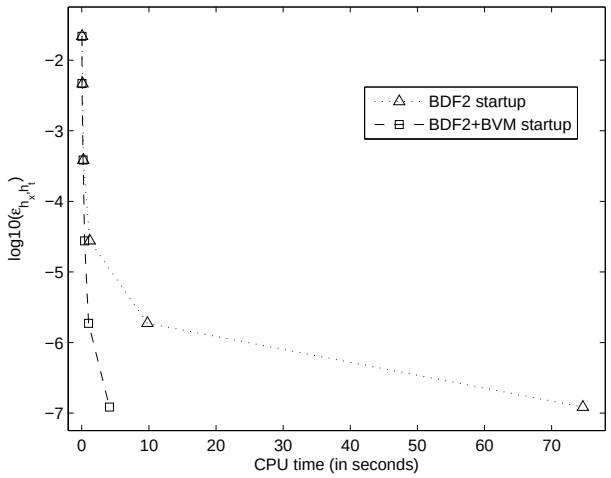

Figure 1. CPU time of BDF2 startup and BDF2+BVM startup versus the error ε_{h_x,h_t}.

We then study the number of iterations of different preconditioners in the right-preconditioned GMRES method. Here we focus on the iteration numbers used to solve the BVM systems on $[h_t, T]$. The resulting linear systems produced from BVMs are solved by the GMRES method with no preconditioner, Strang-type preconditioner [11], and our Crank-Nicolson preconditioner. In Table 2, m, n, n^*, ε_{h_x,h_t} and "order" are defined as before. Let "I_u", "I_s", "I_c" respectively denote the number of iterations of the GMRES method with no preconditioner, Strang-type preconditioner, and our Crank-Nicolson preconditioner. We observe from Table 2 that the Crank-Nicolson preconditioner is very effective. Not only the number of iterations required for convergence does not increase as the mesh size decreases, the GMRES method with Crank-Nicolson preconditioner needs far fewer iterations as well when compared with the Strang-type preconditioner and the unpreconditioned case.

Table 2. Errors and convergence orders of FOCBVM and the iterative numbers of GMRES method respectively with no preconditioner, Strang-type preconditioner and Crank-Nicolson preconditioner in Merton's model.

m	n	n^*	ε_{h_x,h_t}	order	I_u	I_s	I_c
21	5	21	2.19e-2	-	51	32	12
45	10	28	4.66e-3	2.23	111	33	12
101	20	35	3.86e-4	3.59	285	37	11
233	40	42	2.77e-5	3.80	>1000	42	11
481	80	42	1.88e-6	3.88	>1000	55	12
1089	160	49	1.22e-7	3.94	>1000	>60	11

Finally, we examine the performance of the first iteration in the right-preconditioned GMRES method. The residuals are displayed in the columns "None", "Strang" and "Crank-Nicolson", respectively for no preconditioner, Strang-type preconditioner and our Crank-Nicolson preconditioner. From Table 3, the case with Crank-Nicolson preconditioner obtains far better results than the other two in the first iteration. We also observe that the residual steadily decreases as n is doubled. This result theoretically echoes the conclusion in Theorem 4.3 because n is the dominant part of the convergence rate.

Though Theorem 4.3 only guarantees the residual in the first iteration, we are also interested in the behavior of the following iterations. The numerical tests are carried out on the finest mesh with $m = 1089$ and $n = 160$. From Figure 2, we see that the GMRES method with Crank-Nicolson preconditioner still performs well after the first iteration.

Table 3. Residuals of the first iteration in GMRES method with no preconditioner, Strang-type circulant preconditioner, and Crank-Nicolson preconditioner.

m	n	None	Strang	Crank-Nicolson
21	5	5.92e-1	6.11e-1	3.68e-3
45	10	5.93e-1	6.26e-1	2.78e-3
101	20	5.99e-1	6.31e-1	2.14e-3
233	40	5.88e-1	6.79e-1	1.50e-3
481	80	5.28e-1	7.82e-1	8.77e-4
1089	160	5.24e-1	8.79e-1	4.15e-4

6. Concluding remarks

We study the numerical solution of option pricing models under the jump-diffusion process in this paper. We have employed the FOCBVM on a refined mesh to solve

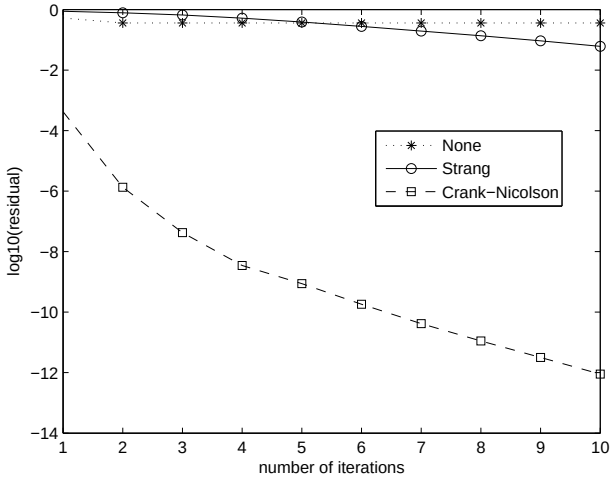

Figure 2. Residuals of first ten iterations in GMRES method for no preconditioner, Strang-type circulant preconditioner and Crank-Nicolson preconditioner with $m = 1089$ and $n = 160$.

the PIDE in order to obtain high order accuracy in spatial and time direction. Compared with [11], the method in this paper gives the following two improvements. First, the direct BDF2 startup in [11] is replaced by the BDF2 and BVM startup used in the initial time layer. Secondly, in the implementation of GMRES method for solving the resulting linear system, the Crank-Nicolson preconditioner substitutes circulant-type preconditioner. Both improvements significantly reduce the operation cost, and thus result in a rapid and highly accurate approach.

Recall that only the residual in the first iteration of the GMRES method with Crank-Nicolson preconditioner is studied in this paper. In fact, numerical results show that this preconditioner still leads to fast convergence for the following iterations. Therefore, the overall iterative convergence of the Crank-Nicolson preconditioner will be investigated for future work. The choice of constant c in our BDF2 startup procedure could also be considered. In addition, we will consider the pricing problem of different kind of options, e.g., American options and exotic options, and extend our method to other financial models such as models with stochastic volatility.

Acknowledgements

This work was partially supported by the research grant 033/2009/A from FDCT of Macao, UL020/08-Y2/MAT/JXQ01/FST and RG063/08-09S/SHW/FST from University of Macau.

References

[1] A. Almendral and C. Oosterlee, *Numerical valuation of options with jumps in the underlying*, Appl. Numer. Math. 53 (2005), pp. 1–18.
[2] D. Bertaccini, *A circulant preconditioner for the systems of LMF-based ODE codes*, SIAM J. Sci. Comput. 22 (2000), pp. 767–786.
[3] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Polit. Economy 81 (1973), pp. 637–654.
[4] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Gordan and Breach, Amsterdam, 1998.
[5] R. Chan, M. Ng, and X. Jin, *Strang-type preconditioners for systems of LMF-based ODE codes*, IMA J. Numer. Anal. 21 (2001), pp. 451–462.
[6] L. Feng and V. Linetsky, *Pricing options in jump-diffusion models: An extrapolation approach*, Oper. Res. 56 (2008), pp. 304–325.

[7] Y. d'Halluin, P. Forsyth, and K. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA J. Numer. Anal. 25 (2005), pp. 87–112.
[8] J. Hou and H. Sun, *A preconditioner from Crank-Nicolson scheme for systems of LMF-based ODE code*, in: Deng Ding, Xiao-Qing Jin, Hai-Wei Sun (Eds.), Recent Advances in Computational Mathematics, Higher Education Press, Beijing & International Press of Boston, Somerville, 2008, pp. 139–149.
[9] S. Kou, *A jump-diffusion model for option pricing*, Manag. Sci. 48 (2002), pp. 1086–1101.
[10] S. Lee and H. Sun, *Fourth order compact scheme with local mesh refinement for option pricing in jump-diffusion model*, preprint.
[11] S. Lee and H. Sun, *Fourth order compact boundary value method for option pricing with jumps*, Adv. Appl. Math. Mech. 1 (2009), pp. 845–861.
[12] R. Merton, *Option pricing when underlying stock returns are discontinuous*, J. Financ. Eco. 3 (1976), pp. 125–144.
[13] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math. 43 (1984), pp. 309–327.
[14] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia, 2000.
[15] D. Tangman, A. Gopaul, and M. Bhuruth, *Numerical pricing of options using high-order compact finite difference schemes*, J. Comput. Appl. Math. 218 (2008), pp. 270–280.
[16] D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Wiley and Sons, New York, 2000.
[17] P. Wilmott, *Derivatives: The Theory and Practice of Financial Engineering*, John Wiley and Sons Ltd., West Sussex, England, 1998.
[18] J. Zhang, H. Sun, and J. Zhao, *High order compact scheme with multigrid local mesh refinement procedure for convection diffusion problems*, Comput. Methods Appl. Mech. Engrg. 191 (2002), pp. 4661–4674.