A Fast High-Order Sinc-Based Algorithm for Pricing Options under Jump-Diffusion Processes [*]

Jun Liu [†] Hai-Wei Sun [‡]

Abstract

An implicit-explicit Euler scheme in temporal direction is employed to discretize a partial integro-differential equation, which arises in pricing options under jump-diffusion process. Then the semi-discretized equation is approximated in space by the Sinc-Galerkin method with exponential accuracy. Meanwhile, the domain decomposition method is incorporated to handle the non-smoothness of the payoff function, and the improved fast Gauss transform is applied to accelerate the evaluation of the jump integral term. An effective preconditioner is proposed for solving the resulting dense Toeplitz-related systems by preconditioned GMRES. Numerical tests are performed to illustrate the efficiency of the proposed algorithm.

Key words: domain decomposition, improved fast Gauss transform, Sinc method, Toeplitz, integro-differential equations
Mathematics Subject Classification: 65T50; 65M06; 65M12; 91B28

1 Introduction

In 1976, Merton in his initiative paper [22] gave a formula for pricing option contracts under the jump-diffusion model to improve the one in Black-Scholes' pure-diffusion model [4]. The jump-diffusion model excels the pure-diffusion model in providing a better fit to underlying asset price dynamics, and generating implied volatility curves that are more conform with real market option prices [5].

Pricing options under the jump-diffusion process leads to a partial integro-differential equation (PIDE) with a non-local integral term. Various numerical pricing techniques have been developed for solving this PIDE. Amin [2] suggested an explicit approach based on multinomial trees, which achieved first order accuracy in time. Andersen and Andreasen [3] proposed a second order accurate alternating direction implicit method in time, with fast Fourier transform (FFT) for evaluating the convolution integral. Almendral and Oosterlee [1] discretized the PIDE in space by second order finite difference and finite elements respectively, and in time by the second order backward differentiation formula. Tavella and Randall [37] employed an implicit scheme which yields dense linear systems

[*] The research was partially supported by the research grants 033/2009/A and 005/2012/A1 from FDCT of Macao, and MYRG206(Y1-L4)-FST11-SHW from University of Macau.

[†] Department of Mathematics, Southern Illinois University, Carbondale, IL, USA (jliu@math.siu.edu).

[‡] Department of Mathematics, University of Macau, Macao, China (HSun@umac.mo).

due to the non-local integral term and then solved the resulting system by a fixed point iterative method. Similar treatments were also discussed in [11]. Any implicit treatment of the integral term will inevitably lead to dense systems, which are expensive to solve [37]. Therefore, Cont and Voltchkova [9] used an implicit-explicit (IMEX) Euler scheme which treated the differential terms implicitly and the integral term explicitly.

These numerical methods described above can achieve at most second order accuracy in space. A natural difficulty in developing a high-order scheme for evaluating various options is the lack of smoothness in payoff functions [21]. To achieve high-order convergence rate, one may need to perform a grid-stretching technique [37], which transforms the original PIDE analytically as in [36] to maintain a uniform mesh. Alternatively, one could directly work on the non-uniform mesh as in [16, 38], which will in return ruin the tridiagonal structure of the differential discretization matrix and give rise to an unstructured dense matrix for the integral term. To keep a favorable structure and high-order accuracy, we resort to the Sinc method [18, 32] for the Toeplitz-related structure of its discretized matrix in approximating the differential terms. But the direct application of the Sinc method to the PIDE is also rendered difficult owing to the non-smoothness of payoff functions. The Sinc method copes with boundary singularities efficiently through a specific conformal map which introduces more grid cells around the boundaries. This property allows the domain decomposition method (DDM) [29] to be specified such that the interior singularities of the original problem become boundary singularities in each sub-domain problem [25], and consequently concentrates more grid cells around the interior singular point as well as the boundaries. The Sinc DDM, which turns out to be of exponential accuracy, was initially presented by Lybeck and Bowers [20] for solving a homogeneous two point boundary value problem (BVP) and later applied to Poisson's equation [19].

Since the IMEX-Euler scheme has only first order accuracy, Feng and Linetsky [13] advised to ameliorate the IMEX scheme by a high order extrapolation approach. Following this idea, we propose to discretize the PIDE in space by the Sinc DDM and in time by the IMEX-Euler scheme facilitated by the Richardson extrapolation [33]. We prefer Richardson extrapolation for its straightforward implementation and robustness in control discretization error in time. Since the resulting linear system discretized by the Sinc DDM still keeps the favorable Toeplitz-related structure, we employ preconditioned GMRES to solve such a system [26]. The integral term is approximated by the Sinc quadrature method [18, 32] with exponential accuracy. Nevertheless, the mesh grids used in the Sinc quadrature method are fully non-uniform, and hence the associated matrix from the integral term loses the Toeplitz structure completely. Inspired by the technique addressed in [6] to speed the Gaussian summations arising from pricing options by fast Gauss transform [14], we then apply the improved fast Gauss transform (IFGT) [23, 39] with linear complexity to accelerate the approximated integral summations.

The paper is organized as follows. In the next section, the mathematical model for pricing options with the jump-diffusion process is given in terms of a PIDE, and the IMEX-Euler scheme with Richardson extrapolation approach is illustrated. The Sinc method for discretizing the semi-discretized BVP is formulated and its stability is studied in Section 3. In Section 4, we investigate the Sinc DDM for the semi-discretized BVP, and then construct a modified tridiagonal preconditioner based on the idea in [26] for solving the resulting linear system. In Section 5, the Sinc quadrature method is used to approximate the integral terms and the IFGT method is employed for speeding up the integral summations. Numerical results are presented in Section 6 to demonstrate the performance of our method. Finally, some conclusions are given in Section 7.

2 PIDE for option pricing and IMEX scheme

Let s be the price of a given underlying asset, which is assumed to follow a jump-diffusion process [8]. The value of a contingent claim that relies on the underlying asset with maturity T and strike price K, denoted by $V(x,t)$, satisfies a PIDE [8]

$$V_t = \frac{1}{2}\sigma^2 V_{xx} + (r - \nu - \lambda\eta - \frac{1}{2}\sigma^2)V_x - (r+\lambda)V + \lambda\int_{-\infty}^{\infty} V(y,t)\varphi(y-x)\mathrm{d}y, \tag{2.1}$$

in $(-\infty,\infty)\times(0,T]$, where $x = \log(s/K)$ is the logarithmic price, t is the time until expiration, $\sigma > 0$ is the stock return volatility, $r \geq 0$ is the risk free interest rate, $\nu \geq 0$ is the continuous dividend yield, η is the expected relative jump size, λ is the jump intensity and $\varphi(x)$ is the probability density function of the distribution of jump sizes.

In this paper, we consider Merton's jump-diffusion model [22], where

$$\varphi(x) = \frac{e^{-(x-\mu)^2/2\gamma^2}}{\sqrt{2\pi}\gamma} \tag{2.2}$$

is the probability density function of a Gaussian distribution with mean μ and variation γ. It is easy to check that $\eta = \int_{-\infty}^{\infty}(e^x - 1)\varphi(x)dx = e^{\mu+\gamma^2/2} - 1$.

2.1 Boundary-initial conditions

For the PIDE (2.1), we denote the differential part of the operator by $\mathcal{P}$ and the integral part by $\mathcal{Q}$. Then (2.1) reads

$$V_t = \mathcal{P}(V) + \lambda\mathcal{Q}(V), \tag{2.3}$$

where

$$\mathcal{P}(V) = \frac{1}{2}\sigma^2 V_{xx} + (r - \nu - \lambda\eta - \frac{1}{2}\sigma^2)V_x - (r+\lambda)V \tag{2.4}$$

and

$$\mathcal{Q}(V) = \int_{-\infty}^{\infty} V(y,t)\varphi(y-x)\mathrm{d}y. \tag{2.5}$$

There are various types of boundary and initial conditions for (2.1) depending on the type of options. The initial and asymptotic boundary conditions are given by [11]

$$\begin{cases} V(x,0) = K(e^x - 1)^+, & \\ V(x,t) \to 0, & \text{as} \quad x \to -\infty, \\ V(x,t) \to Ke^{x-\nu t} - Ke^{-rt}, & \text{as} \quad x \to +\infty, \end{cases} \tag{2.6}$$

for European call options.

In the case of standard double-barrier knock-out call options without rebate, the contract becomes worthless if either of the barriers is reached before the option expiry date T.

Let x_u and x_d be the up and down barrier levels, respectively. Then the initial condition remains the same as in (2.6), while the boundary conditions vanish outside the barriered domain (x_d, x_u), that is [13],

$$\begin{cases} V(x,0) = K(e^x - 1)^+, & \\ V(x,t) = 0, & \text{if} \quad x \leq x_d, \\ V(x,t) = 0, & \text{if} \quad x \geq x_u. \end{cases} \tag{2.7}$$

It is well-known that barrier options are more challenging for traditional numerical pricing methods, which often display very poor convergence [37]. We will show that our method is equally effective for this kind of path-dependent options.

In order to numerically solve the initial boundary value problem (2.3 & 2.6) for European call options, we first truncate the spatial domain $\mathbb{R} = (-\infty, \infty)$ to a finite interval $\mathbb{X} = (x_{\min}, x_{\max})$, and then impose the asymptotic boundary conditions (2.6) on the artificial boundary points $x_{\min}$ and $x_{\max}$ [11], i.e.,

$$\begin{cases} V(x_{\min}, t) = 0, \\ V(x_{\max}, t) = Ke^{x_{\max} - \nu t} - Ke^{-rt}. \end{cases} \tag{2.8}$$

On the other hand, the initial boundary value problem (2.3 & 2.7) for double barrier options already has a finite spatial domain (x_d, x_u) and hence we choose $\mathbb{X} = (x_d, x_u)$.

2.2 Splitting of the integral term

The infinite integral domain $\mathbb{R}$ is splitted into $\mathbb{X}$ and $\overline{\mathbb{X}} = \mathbb{R} \setminus \mathbb{X}$, and the integral term in (2.5) can be decomposed as in [1],

$$\begin{aligned} \mathcal{Q}(V) &= \int_{-\infty}^{\infty} V(y,t)\varphi(y-x)\mathrm{d}y \\ &= \int_{\mathbb{X}} V(y,t)\varphi(y-x)\mathrm{d}y + \int_{\overline{\mathbb{X}}} V(y,t)\varphi(y-x)\mathrm{d}y \\ &\equiv \mathcal{Q}_{\mathbb{X}}(V) + \mathcal{Q}_{\overline{\mathbb{X}}}(V). \end{aligned} \tag{2.9}$$

For European call options, we note that $V(x,t)$ is approximated by the asymptotic boundary conditions (2.6) on $\overline{\mathbb{X}}$ as follows,

$$\begin{cases} V(x,t) = 0, & x \leq x_{\min}, \\ V(x,t) = Ke^{x-\nu t} - Ke^{-rt}, & x \geq x_{\max}. \end{cases} \tag{2.10}$$

By the expression of $\varphi(z)$ in (2.2), we have the approximation [35]

$$\begin{aligned} \mathcal{Q}_{\overline{\mathbb{X}}}(V) &\approx \int_{x_{\max}}^{\infty} (Ke^{y-\nu t} - Ke^{-rt})\varphi(y-x)\mathrm{d}y \\ &= Ke^{x-\nu t+\mu+\gamma^2/2}\Phi(\frac{x - x_{\max} + \mu + \gamma^2}{\gamma}) - Ke^{-rt}\Phi(\frac{x - x_{\max} + \mu}{\gamma}), \end{aligned}$$

where

$$\Phi(x) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-z^2}{2}} dz \tag{2.11}$$

is the standard cumulative normal distribution. We remark that for double-barrier options, the second term $\mathcal{Q}_{\overline{\mathbb{X}}}(V)$ always equals to zero; see (2.7). The remaining first term $\mathcal{Q}_{\mathbb{X}}(V)$ will be approximated by the Sinc quadrature method, which will be discussed in Section 5.1.

2.3 IMEX-Euler Scheme and Extrapolation

Since the numerical integration of $\mathcal{Q}_{\mathbb{X}}(V)$ leads to a dense matrix, we utilize the IMEX-Euler scheme [13] to avoid inverting the dense matrix. The discretization in temporal direction is performed using a uniform grids. Let the number of time grids to be J, and the time grids are given by

$$t_j = j\Delta t, \qquad j = 0, 1, \dots, J,$$

with time step $\Delta t = T/J$. Let $V^j = V(x, t_j)$, $j = 0, 1, \dots, J$. The IMEX-Euler scheme starts with the initial condition $V^0 = V(x, 0)$ and marches forward according to

$$\frac{V^{j+1} - V^j}{\Delta t} = \mathcal{P}(V^{j+1}) + \lambda \mathcal{Q}(V^j), \quad j = 0, 1, \dots, J-1.$$

Hence

$$V_{xx}^{j+1} + p(x) V_x^{j+1} + q(x) V^{j+1} = f^j(x), \quad j = 0, 1, \dots, J-1, \tag{2.12}$$

where

$$p(x) = \frac{2}{\sigma^2}\left(r - \nu - \lambda\eta - \frac{\sigma^2}{2}\right), \quad q(x) = -\frac{2}{\sigma^2}\left(r + \lambda + \frac{1}{\Delta t}\right),$$

and

$$f^j(x) = -\frac{2}{\Delta t \sigma^2}\left(V^j + \lambda \Delta t \mathcal{Q}(V^j)\right), \quad j = 0, 1, \dots, J-1. \tag{2.13}$$

The solution V^{j+1} is subject to the following boundary conditions for European call options,

$$\begin{cases} V^{j+1}(x_{\min}) = 0, \\ V^{j+1}(x_{\max}) = K e^{x_{\max} - \nu(j+1)\Delta t} - K e^{-r(j+1)\Delta t}, \end{cases} \tag{2.14}$$

and the homogeneous boundary conditions for double-barrier call options,

$$\begin{cases} V^{j+1}(x_{\min}) = 0, \\ V^{j+1}(x_{\max}) = 0. \end{cases} \tag{2.15}$$

At the $(j+1)$th time step, the BVP (2.12 & 2.14) or (2.12 & 2.15) is solved to approximate V^{j+1} on $\mathbb{X}$. However, the depicted IMEX-Euler scheme is only first order accurate [11]. Suppose that the spatial discretization based on the IMEX-Euler scheme is unconditionally stable, the Richardson extrapolation method could be applied to achieve high order convergence in temporal direction.

Let $V_{k,1}^J$ be the solution obtained at T with step size $\Delta t / 2^{k-1}$, $k = 1, \dots, S$, where S is the intended stage number of extrapolation. Then the Richardson extrapolation formula is given by

$$V_{k,l}^J = \frac{2^{l-1} V_{k,l-1}^J - V_{k-1,l-1}^J}{2^{l-1} - 1}, \quad k = 2, \dots, S, \quad l = 2, \dots, k, \tag{2.16}$$

where $V_{k,l}^J$ denote intermediate approximations with step size $\Delta t/2^{k-1}$ after l extrapolation stages. By this recursion, a more accurate approximation $V_{S,S}^J$ to $V(x,T)$ is obtained at the Sth stage. In total, it costs $m = (2^S - 1)J$ time steps to accomplish above extrapolation.

3 Sinc discretization for BVP

3.1 Sinc method

In this subsection, we will introduce the symmetric Sinc-Galerkin method only for the BVP associated with (2.12) and the homogeneous boundary conditions (2.15). The non-homogeneous boundary conditions (2.14) can be transformed to a homogeneous one by a simple transformation of the solution [18, p159],

$$\widehat{V}(x,t) = V(x,t) - V(x_{\min},t)\frac{x_{\max}-x}{x_{\max}-x_{\min}} - V(x_{\max},t)\frac{x-x_{\min}}{x_{\max}-x_{\min}}. \tag{3.17}$$

Thus, we turn to focus on the following linear two-point BVP

$$\begin{cases} \mathcal{L}u(x) \equiv u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad x \in \mathbb{X}, \\ u(x_{\min}) = 0, \quad u(x_{\max}) = 0. \end{cases} \tag{3.18}$$

By the adjective "symmetric" we mean the derived linear system is symmetric whenever $p(x) \equiv 0$; see [32, Section 7.2.7] for more details.

The Sinc function is given by

$$\mathrm{Sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0, \\ 1, & x = 0. \end{cases}$$

In order to approximate a function on the bounded domain $\mathbb{X} = (x_{\min}, x_{\max})$, we define a conformal map [18, 32]

$$\phi(x) = \ln\left(\frac{x - x_{\min}}{x_{\max} - x}\right) \tag{3.19}$$

from the bounded interval $\mathbb{X}$ onto the whole real line $\mathbb{R}$, such that $\phi(x_{\min}) = -\infty$ and $\phi(x_{\max}) = \infty$. Compounding with this map we could approximate the solution $u(x)$ by a truncated expansion [32]

$$u(x) \approx U_{\mathbb{X}}(x) \equiv \sum_{k=-N}^{N} u_k S_k^h \circ \phi(x), \quad n = 2N+1, \tag{3.20}$$

where

$$S_k^h \circ \phi(x) \equiv \frac{\mathrm{Sinc}\left(\frac{\phi(x)-kh}{h}\right)}{\sqrt{\phi'(x)}}, \qquad (h > 0),$$

are basis functions and $\{u_k\}_{k=-N}^{N}$ are coefficients to be determined. Moreover, there holds

$$U_{\mathbb{X}}(x_k) = \frac{u_k}{\sqrt{\phi'(x_k)}} \tag{3.21}$$

at Sinc grid points $x_k = \phi^{-1}(kh), k = -N, \dots, N$.

The symmetric Sinc-Galerkin method enables us to get the coefficients $\{u_k\}_{k=-N}^{N}$ through solving a joint linear system of equations, which is formulated by combining the system derived from approximating the inner products of the standard Galerkin method, i.e.,

$$\langle \mathcal{L}U_{\mathbb{X}} - f, S_k^h \circ \phi \rangle \equiv \int_{\mathbb{X}} (\mathcal{L}U_{\mathbb{X}}(x) - f(x)) S_k^h \circ \phi(x) \mathrm{d}x = 0, \quad -N \le k \le N,$$

and the system obtained by the collocation method, i.e.,

$$\mathcal{L}U_{\mathbb{X}}(x_k) = f(x_k), \quad -N \le k \le N.$$

Suppose $\mathbb{S}$ is a simply-connected domain in the complex plane with boundary points $x_{\min} \neq x_{\max}$, such as $\mathbb{X}$ and so forth. The family of analytic functions g in $\mathbb{S}$ is denoted by $\mathscr{H}(\mathbb{S})$. We define two function spaces

$$\mathscr{H}^{\infty}(\mathbb{S}) = \left\{ g \in \mathscr{H}(\mathbb{S}) : \quad \sup_{z \in \mathbb{S}} |g(z)| < \infty \right\}$$

and

$$\mathscr{L}_{\zeta}(\mathbb{S}) = \left\{ g \in \mathscr{H}(\mathbb{S}) : \exists\, c_1 > 0, \mathbf{s.t.}, \forall z \in \mathbb{S}, |f(z)| \le c_1 \frac{|e^{\phi(z)}|^{\zeta}}{(1 + |e^{\phi(z)}|)^{2\zeta}} \right\}, \quad \zeta > 0.$$

The following assumption on functions p, q and f are required to assure the exponential convergence of the symmetric Sinc-Galerkin method.

Assumption 1 [32, p489] *The BVP* (3.18) *is assumed to satisfy*

(A1) *The equation has a unique solution u with $\sqrt{\phi'}u \in \mathscr{L}_{\zeta}(\mathbb{X})$;*

(A2) *Functions $\frac{p}{\phi'}$, $\frac{1}{\phi'}(\frac{p}{\phi'})'$, $\frac{q}{(\phi')^2}$, $(\frac{1}{\phi'})'$ and $\frac{1}{\phi'}(\frac{1}{\phi'})''$ belong to $\mathscr{H}^{\infty}(\mathbb{X})$, and $\frac{f}{(\phi')^2} \in \mathscr{L}_{\zeta}(\mathbb{X})$;*

(A3) *Re$\left\{ \left[\frac{1}{2\phi'}(\frac{1}{\phi'})'' - \frac{1}{4}[(\frac{1}{\phi'})']^2 + \frac{2q-p'}{2(\phi')^2} \right] \right\} \le 0$ for $a < x < b$, where Re(z) is the real part of z, and $\frac{1}{\phi'}$ is uniformly bounded in $\mathbb{X}$.*

Let

$$\delta_{ik}^{(1)} \equiv \delta^{(1)}(k-i) = \left\{ \begin{array}{cc} 0, & k = i, \\ \frac{(-1)^{k-i}}{k-i}, & k \neq i, \end{array} \right.$$

and

$$\delta_{ik}^{(2)} \equiv \delta^{(2)}(k-i) = \left\{ \begin{array}{cc} \frac{-\pi^2}{3}, & k = i, \\ \frac{2(-1)^{k-i}}{(k-i)^2}, & k \neq i. \end{array} \right.$$

Define two n-by-n matrices T_1 and T_2 with

$$[T_1]_{i,k} = \left[\delta_{ik}^{(1)} \right]$$

and

$$[T_2]_{i,k} = \left[\delta_{ik}^{(2)} \right]$$

respectively. We remark that an n-by-n matrix M is said to be Toeplitz if $[M]_{i,k} = m_{i-k}$ [7], and therefore both T_1 and T_2 are Toeplitz. For any function ψ on $\mathbb{X}$, we define the corresponding diagonal matrix

$$\mathcal{D}[\psi] \equiv \text{diag } [\psi(x_{-N}), \ldots, \psi(x_N)].$$

The following theorem on the symmetric Sinc-Galerkin approximate solution (3.20) was established in [32].

Theorem 1 [32, p490] *Let* Assumption 1 *be satisfied,* $d = \frac{\pi}{2}$, $h = (\frac{\pi d}{\zeta N})^{1/2}$, *and the energy norm of a given function* $g(x)$ *on* $\mathbb{X}$ *is defined by*

$$\|g\|_e = \left[\int_{\mathbb{X}} (|g(x)|^2 + |g'(x)|^2)\mathrm{d}x\right]^{1/2}.$$

Let $\boldsymbol{u} = [u_{-N}, \ldots, u_N]^\intercal$ *be the solution of the following Sinc-Galerkin system*

$$G_\phi \boldsymbol{u} = h^2 \mathcal{D}[(\phi')^{-3/2}]\boldsymbol{f}, \tag{3.22}$$

where

$$G_\phi \equiv \left(T_2 - \frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1]) + h^2\mathcal{D}[\psi_2]\right), \tag{3.23}$$

with $\psi_1 = \frac{p}{\phi'}$ *and* $\psi_2 = \left[\frac{1}{2\phi'}(\frac{1}{\phi'})'' - \frac{1}{4}[(\frac{1}{\phi'})']^2 + \frac{2q-p'}{2(\phi')^2}\right]$, *and* $\boldsymbol{f} = [f(x_{-N}), \ldots, f(x_N)]^\intercal$, *then there exists a constant* c_2 *independent of* N, *such that*

$$\|u - U_{\mathbb{X}}\|_e \le c_2 N^{5/2} \exp(-(\pi d \zeta N)^{1/2}),$$

where $u(x)$ *is the true solution of* (3.18).

What distinguishes the Sinc method from conventional ones is that it achieves exponential accuracy by solving a linear system with Toeplitz-related structure. Given an n-by-1 vector $\boldsymbol{y}$, the product $T_1\boldsymbol{y}$ and $T_2\boldsymbol{y}$ are calculated in $\mathcal{O}(n \log n)$ operations by FFT [7, 27], and $\mathcal{D}[\psi]\boldsymbol{y}$ in $\mathcal{O}(n)$ operations. In total, the matrix-vector product $G_\phi \boldsymbol{y}$ is computed in $\mathcal{O}(n \log n)$ operations. Furthermore, several fast iterative solvers based on matrix-vector products for solving the Sinc-Galerkin system (3.22) have been studied. They include the preconditioned conjugate gradient (PCG) method for the symmetric case [26], the preconditioned conjugate gradient method for normal equations (CGNE) and the preconditioned GMRES [30] for the non-symmetric case [28]. Finally, we point out that truncating at the Nth term from both sides in (3.20) is only for simplicity, and a wiser truncation should be selected according to the characteristics of pertaining functions [18].

3.2 Stability analysis

We now study the stability of the symmetric Sinc-Galerkin scheme. Without losing generality, we deal with double-barrier options due to their homogeneous boundaries and natural bounded integral domain. At the jth time step, we denote the option prices at all the spatial grid points by

$$\boldsymbol{w}^j = [V^j(x_{-N}), \ldots, V^j(x_N)]^\intercal.$$

According to the Sinc quadrature method, which will be stated in Section 5.1, the term $\mathcal{Q}(V^j)$ is approximated by

$$\mathcal{Q}(V^j)(x_k) = \int_{\mathbb{X}} V^j(y)\varphi(y-x_k)\mathrm{d}y \approx \sum_{l=-N}^{N} \frac{h\varphi(x_l - x_k)}{\phi'(x_l)} V^j(x_l)$$

at every spatial grid point $x_k = \phi^{-1}(kh), k = -N, \ldots, N$, where we use the fact that $\mathcal{Q}_{\overline{\mathbb{X}}}(V^j)(x_k) = 0$. Therefore, the integral terms $\mathcal{Q}(V^j)$ at all spatial grids x_k are evaluated by one matrix-vector product $Q\boldsymbol{w}^j$, where Q is an n-by-n unstructured dense matrix with

$$[Q]_{k,l} = \frac{h\varphi(x_l - x_k)}{\phi'(x_l)}.$$

Substituting the expression (2.13) into the Sinc-Galerkin system (3.22) and noting the relation (3.21), we determine $\boldsymbol{w}^{j+1}$ by solving

$$G_\phi \mathcal{D}[(\phi')^{1/2}]\boldsymbol{w}^{j+1} = -\frac{2h^2}{\Delta t \sigma^2}\mathcal{D}[(\phi')^{-3/2}](\boldsymbol{w}^j + \lambda \Delta t Q \boldsymbol{w}^j)$$

according to the IMEX-Euler scheme. By further letting $\kappa = -\frac{2h^2}{\Delta t\sigma^2}$ and

$$A = \kappa \mathcal{D}[(\phi')^{-1/2}] G_\phi^{-1} \mathcal{D}[(\phi')^{-3/2}],$$

we obtain

$$\boldsymbol{w}^{j+1} = (A + \lambda \Delta t A Q)\boldsymbol{w}^j. \tag{3.24}$$

We give the following definition and lemma required for stating our main theorem.

Definition 1 [17, p189] *Let $\|\cdot\|$ denote a matrix norm. The scheme* (3.24) *is said to be Lax–Richtmyer stable if, for any $T > 0$, there is a constant $c_3 > 0$ such that*

$$\|(A + \lambda \Delta t A Q)^{j+1}\| \le c_3$$

for all time step $\Delta t > 0$ and integer j for which $(j+1)\Delta t \le T$.

Denote the eigenvalues of an n-by-n matrix M by $\theta_i(M), i = 1, 2, \ldots, n$. The spectral radius of M is defined by

$$\rho(M) = \max_{1 \le i \le n} |\theta_i(M)|.$$

Lemma 1 [15, p297] *If $\rho(M) < 1$, there exists a matrix norm $\|\cdot\|_\rho$ such that $\|M\|_\rho < 1$.*

In the following theorem, we prove that the spectral radius of the matrix A in (3.24) is always strictly less than one, which will be employed to prove the stability of our scheme.

Theorem 2 *Given constants $r \ge 0$, $\lambda \ge 0$ and $\sigma > 0$, then $\rho(A) < 1$ in* (3.24).

Proof: Assume that $(\theta, \widehat{\boldsymbol{z}})$ is an eigenpair of A with $\widehat{\boldsymbol{z}} \neq 0$. Let $\boldsymbol{z} = \mathcal{D}[(\phi')^{1/2}]\widehat{\boldsymbol{z}}$, then

$$\theta G_\phi \boldsymbol{z} = \kappa \mathcal{D}[(\phi')^{-2}]\boldsymbol{z}.$$

With $p(x)$ and $q(x)$ defined in (2.12) and G_ϕ in (3.23), we have

$$\theta\left(T_2 - \frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1]) + h^2\mathcal{D}[\psi_2]\right)\boldsymbol{z} = \kappa\mathcal{D}[(\phi')^{-2}]\boldsymbol{z}, \tag{3.25}$$

where

$$\psi_1 = \frac{2(r-\nu-\lambda\eta-\frac{\sigma^2}{2})}{\phi'\sigma^2}$$

and

$$\psi_2 = \left\{\frac{1}{2\phi'}\left(\frac{1}{\phi'}\right)'' - \frac{1}{4}\left[\left(\frac{1}{\phi'}\right)'\right]^2 - \frac{2(r+\lambda+\frac{1}{\Delta t})}{(\phi')^2\sigma^2}\right\} < 0.$$

By Theorem 4.18 in [18, p151], T_2 is symmetric negative definite and T_1 is skew-symmetric. For the given conformal map $\phi(x)$ in (3.19), by straightforward calculation we have $\frac{1}{\phi'} > 0$ and $(\frac{1}{\phi'})'' < 0$, thus $\mathcal{D}[\psi_2]$ is also negative definite.

Multiplying (3.25) from the left by $\boldsymbol{z}^*$, where $\boldsymbol{z}^*$ is the conjugate transpose of $\boldsymbol{z}$, leads to

$$\theta\boldsymbol{z}^*\left(T_2 - \frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1]) + h^2\mathcal{D}[\psi_2]\right)\boldsymbol{z} = \kappa\boldsymbol{z}^*\mathcal{D}[(\phi')^{-2}]\boldsymbol{z}. \tag{3.26}$$

Let $\widehat{\psi_2} = \frac{1}{2\phi'}(\frac{1}{\phi'})'' - \frac{1}{4}[(\frac{1}{\phi'})']^2 - \frac{2(r+\lambda)}{(\phi')^2\sigma^2}$, then $\widehat{\psi_2} < 0$ and

$$\begin{aligned} h^2\mathcal{D}[\psi_2] &= h^2\mathcal{D}\left\{\frac{1}{2\phi'}\left(\frac{1}{\phi'}\right)'' - \frac{1}{4}\left[\left(\frac{1}{\phi'}\right)'\right]^2 - \frac{2(r+\lambda)}{(\phi')^2\sigma^2}\right\} - h^2\mathcal{D}\left[\frac{2}{\Delta t\sigma^2(\phi')^2}\right] \\ &= h^2\mathcal{D}\left[\widehat{\psi_2}\right] + \kappa\mathcal{D}\left[(\phi')^{-2}\right]. \end{aligned}$$

Substituting the above formula into equation (3.26) and recalling that $\kappa = -\frac{2h^2}{\Delta t\sigma^2} < 0$, we obtain

$$\theta = \frac{\kappa\boldsymbol{z}^*\mathcal{D}[(\phi')^{-2}]\boldsymbol{z}}{\boldsymbol{z}^*\left(T_2 + h^2\mathcal{D}[\widehat{\psi_2}]\right)\boldsymbol{z} + \kappa\boldsymbol{z}^*\mathcal{D}[(\phi')^{-2}]\boldsymbol{z} - \boldsymbol{z}^*\left(\frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1])\right)\boldsymbol{z}}, \tag{3.27}$$

where $T_2 + h^2\mathcal{D}[\widehat{\psi_2}]$ is also symmetric negative definite and $\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1]$ is also skew-symmetric. We already know

$$\boldsymbol{z}^*\left(\frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1])\right)\boldsymbol{z}$$

is pure imaginary. Thus according to (3.27) it follows

$$|\theta| \le \frac{|\kappa|\boldsymbol{z}^*\mathcal{D}[(\phi')^{-2}]\boldsymbol{z}}{\left|\boldsymbol{z}^*\left(T_2 + h^2\mathcal{D}[\widehat{\psi_2}]\right)\boldsymbol{z}\right| + |\kappa|\boldsymbol{z}^*\mathcal{D}[(\phi')^{-2}]\boldsymbol{z}} < 1, \tag{3.28}$$

and the result is concluded. □

Next, we use the matrix approach to prove the unconditional stability of the scheme (3.24) in pure-diffusion case without jump (with $\lambda = 0$). The same technique for proving stability was also used in [40].

Theorem 3 *If $\lambda = 0$, then the scheme* (3.24) *is unconditionally stable.*

Proof: By Lemma 1 and Theorem 2, there exists a matrix norm $\|\cdot\|_\rho$ such that $\|A\|_\rho < 1$. Therefore

$$\|A^{j+1}\|_\rho \leq \|A\|_\rho^{j+1} < 1, \quad j = 1, 2 \ldots, J.$$

It follows from Definition 1 that the scheme (3.24) is stable in terms of $\|\cdot\|_\rho$. □

Remark: For the general jump case with $\lambda \neq 0$, we are unable to prove the unconditional stability of the scheme (3.24) due to the quadrature term. However, we do observe the unconditional stability in all of our numerical simulations; see Section 6.

4 Sinc DDM system

4.1 Sinc DDM for BVP

The error brought by the interior non-smooth point of the payoff function will wreck the accuracy of the Sinc approximation. Morlet [25] suggested to handle the interior singularities of the original problem by performing a non-overlapping domain decomposition at the interior point exactly. We refer readers to [20, 25, 24] for details on the formulation and exponential convergence theory of the Sinc DDM.

Assume $\xi \in \mathbb{X} = (x_{\min}, x_{\max})$ is an interior non-smooth point of the right-hand forcing term $f(x)$. Upon splitting the original domain $\mathbb{X}$ into two non-overlapping sub-domains $\mathbb{X}_1 = (x_{\min}, \xi)$ and $\mathbb{X}_2 = (\xi, x_{\max})$, the BVP (3.18) is equivalent to the following two coupled equations

$$\begin{cases} \mathcal{L}u(x) = u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad x \in \mathbb{X}_1, \\ \hfill u(x_{\min}) = 0, \end{cases} \tag{4.29}$$

$$\begin{cases} \mathcal{L}v(x) = v''(x) + p(x)v'(x) + q(x)v(x) = f(x), \quad x \in \mathbb{X}_2 \\ \hfill v(x_{\max}) = 0, \end{cases} \tag{4.30}$$

on u and v respectively. In addition, u and v satisfy the following interface matching conditions [20]

$$u(\xi) = v(\xi), \tag{4.31a}$$

$$u'(\xi) = v'(\xi). \tag{4.31b}$$

Let ϕ_1 and ϕ_2 denote the conformal map (3.19) redefined on $\mathbb{X}_1$ and $\mathbb{X}_2$, respectively. Analogous to (3.20), both solutions of (4.29) on $\mathbb{X}_1$ and of (4.30) on $\mathbb{X}_2$ are approximated by

$$u(x) \approx U_{\mathbb{X}_1}(x) \equiv \sum_{k=-N}^{N} u_k S_k^h \circ \phi_1(x) + u_{N+1}\alpha(x), \quad x \in \mathbb{X}_1, \tag{4.32}$$

and

$$v(x) \approx U_{\mathbb{X}_2}(x) \equiv v_{-N-1}\beta(x) + \sum_{k=-N}^{N} v_k S_k^h \circ \phi_2(x), \quad x \in \mathbb{X}_2, \tag{4.33}$$

respectively, where the extra basis functions $\alpha(x)$ and $\beta(x)$ are given by [20]

$$\alpha(x) = (x - x_{\min})^3 \left[\frac{-3}{(\xi - x_{\min})^4} x + \frac{4\xi - x_{\min}}{(\xi - x_{\min})^4} \right]$$

and

$$\beta(x) = (x - x_{\max})^3 \left[\frac{-3}{(\xi - x_{\max})^4} x + \frac{4\xi - x_{\max}}{(\xi - x_{\max})^4} \right].$$

We also define Sinc grid points $x_k^{(1)}$ and $x_k^{(2)}$ in $\mathbb{X}_1$ and $\mathbb{X}_2$ respectively, where

$$x_k^{(1)} = \phi_1^{-1}(kh) \qquad \text{and} \qquad x_k^{(2)} = \phi_2^{-1}(kh), \qquad k = -N, \dots, N.$$

From the interface condition (4.31a) we obtain

$$u_{N+1} = v_{-N-1},$$

while the second smoothing condition (4.31b) should be approximated by finite difference method [20]. The similar technique was also used in [10] when applying finite difference with DDM to solve the heat equation. Because ξ is a shared boundary point of $\mathbb{X}_1$ and $\mathbb{X}_2$, we prefer to utilize one-sided difference formulas for approximating the first derivative. Given a small $\delta > 0$, we choose the following fourth-order accurate equi-spaced one-sided finite difference scheme for approximating $u'(\xi)$, that is,

$$u'(\xi) \approx \frac{1}{12\delta} [3u(\xi - 4\delta) - 16u(\xi - 3\delta) + 36u(\xi - 2\delta) - 48u(\xi - \delta) + 25u(\xi)] \equiv \mathcal{B}_\delta(u)(\xi).$$

The finite difference approximation of the smoothing condition (4.31b) yields

$$\boldsymbol{b}_1^\intercal \boldsymbol{u} + \alpha_\delta u_{N+1} = -\beta_\delta v_{-N-1} - \boldsymbol{b}_2^\intercal \boldsymbol{v},$$

where

$$\boldsymbol{u} = [u_{-N}, \dots, u_N]^\intercal, \quad \boldsymbol{v} = [v_{-N}, \dots, v_N]^\intercal,$$
$$\alpha_\delta = \mathcal{B}_\delta(\alpha)(\xi), \quad \beta_\delta = \mathcal{B}_{-\delta}(\beta)(\xi),$$
$$\boldsymbol{b}_1 = [\mathcal{B}_\delta(S_{-N} \circ \phi_1)(\xi), \dots, \mathcal{B}_\delta(S_N \circ \phi_1)(\xi)]^\intercal,$$

and

$$\boldsymbol{b}_2 = [\mathcal{B}_{-\delta}(S_{-N} \circ \phi_2)(\xi), \dots, \mathcal{B}_{-\delta}(S_N \circ \phi_2)(\xi)]^\intercal.$$

Here we define $S_k \circ \phi_1(\xi) \equiv 0$ and $S_k \circ \phi_2(\xi) \equiv 0$, $k = -N, \dots, N$.

Imitating the procedure of symmetric Sinc-Galerkin method as in Section 3.1 on $\mathbb{X}_1$ and $\mathbb{X}_2$ respectively, we obtain a coupled Sinc DDM system with 4-by-4 block structure

$$\begin{bmatrix} G_{\phi_1} & 0 & \boldsymbol{c}_\alpha & 0 \\ 0 & G_{\phi_2} & 0 & \boldsymbol{c}_\beta \\ \boldsymbol{b}_1^\intercal & \boldsymbol{b}_2^\intercal & \alpha_\delta & \beta_\delta \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ u_{N+1} \\ v_{-N-1} \end{bmatrix} = \begin{bmatrix} \mathcal{D}[(\phi_1')^{-3/2}] \boldsymbol{f_1} \\ \mathcal{D}[(\phi_2')^{-3/2}] \boldsymbol{f_2} \\ 0 \\ 0 \end{bmatrix}, \tag{4.34}$$

where

$$\boldsymbol{c}_\alpha = \mathcal{D}[(\phi_1')^{-3/2}] \cdot [\mathcal{L}(\alpha(x_{-N}^{(1)})), ..., \mathcal{L}(\alpha(x_N^{(1)}))]^\intercal,$$

$$\boldsymbol{c}_\beta = \mathcal{D}[(\phi_2')^{-3/2}] \cdot [\mathcal{L}(\beta(x_{-N}^{(2)})), ..., \mathcal{L}(\beta(x_N^{(2)}))]^\intercal,$$

$$\boldsymbol{f_l} = [f(x_{-N}^{(l)}), \dots, f(x_N^{(l)})]^\intercal, \, l = 1, 2,$$

and both matrices G_{ϕ_1} and G_{ϕ_2} are defined as in (3.23) with ϕ replaced by ϕ_1 and ϕ_2, respectively. Notice that $\mathbb{X}_1$ and $\mathbb{X}_2$ are non-overlapping, so the whole approximate solution to (3.18) consists of u on $\mathbb{X}_1$ and v on $\mathbb{X}_2$, and the value at the strike point ξ can be taken as either $u(\xi)$ or $v(\xi)$, whose difference in fact is almost zero within the order of the method.

4.2 Preconditioner for Sinc DDM system

We firstly consider the Sinc Galerkin system (3.22) with the matrix G_ϕ given in (3.23)

$$G_\phi \equiv \left(T_2 - \frac{h}{2}(\mathcal{D}[\psi_1]T_1 + T_1\mathcal{D}[\psi_1]) + h^2\mathcal{D}[\psi_2] \right).$$

In order to solve (3.22), Ng and Potts [28] suggested to exploit the GMRES method with an efficient tridiagonal preconditioner

$$P_\phi \equiv Q_1 - \frac{h}{2}\left(\mathcal{D}[\psi_1]Q_2 + Q_2\mathcal{D}[\psi_1]\right) + h^2\mathcal{D}[\psi_2], \tag{4.35}$$

where Q_1 and Q_2 are the tridiagonal Toeplitz matrices

$$Q_1 = \text{tridiag}[1, -2, 1] \quad \text{and} \quad Q_2 = \text{tridiag}[0.5, 0, -0.5];$$

see [28] for more details.

In the following, we extend [28]'s idea to conceive a preconditioner for solving the system (4.34). The coefficient matrix in (4.34) is written as a 2-by-2 block matrix,

$$M = \begin{bmatrix} G & E \\ F & R \end{bmatrix} \equiv \left[\begin{array}{cc|cc} G_{\phi_1} & 0 & \boldsymbol{c}_\alpha & 0 \\ 0 & G_{\phi_2} & 0 & \boldsymbol{c}_\beta \\ \hline \boldsymbol{b}_1^\intercal & \boldsymbol{b}_2^\intercal & \alpha_\delta & \beta_\delta \\ 0 & 0 & 1 & -1 \end{array}\right]. \tag{4.36}$$

We note that R is a 2-by-2 matrix. Therefore, we can construct a preconditioner for (4.36) as

$$P = \begin{bmatrix} \widehat{P} & E \\ 0 & I_2 \end{bmatrix}, \tag{4.37}$$

where $\widehat{P} = \text{diag}[P_{\phi_1}, P_{\phi_2}]$ with the preconditioners P_{ϕ_1} and P_{ϕ_2} corresponding to the matrices G_{ϕ_1} and G_{ϕ_2} as (4.35) respectively, and I_2 is the 2-by-2 identity matrix. Numerical results in Section 6 do demonstrate the efficiency of this preconditioner.

5 Evaluation to the integrals

5.1 Sinc quadrature method

We first give the following Sinc quadrature method with exponential accuracy.

Theorem 4 [18, p70] *Let $d = \frac{\pi}{2}, h = (\frac{\pi d}{\zeta N})^{\frac{1}{2}}$. If $\frac{g}{\phi'} \in \mathscr{L}_\zeta(\mathbb{X})$, then*

$$\left| \int_{\mathbb{X}} g(x) \mathrm{d}x - h \sum_{k=-N}^{N} \frac{g(x_k)}{\phi'(x_k)} \right| = \mathcal{O}\left(\exp(-(\pi d \zeta N)^{1/2}) \right) \tag{5.38}$$

where $x_k = \phi^{-1}(kh), k = -N, \ldots, N$ are the Sinc grid points.

By the Sinc DDM discretization of BVP (2.12 & 2.14) or (2.12 & 2.15), we need to evaluate the integral term $\mathcal{Q}_{\mathbb{X}}(V^j)$ at two set of Sinc grid points, $\mathbb{G}_1$ and $\mathbb{G}_2$, where

$$\mathbb{G}_1 = \{x_k^{(1)} = \phi_1^{-1}(kh) : k = -N, \ldots, N\} \subset \mathbb{X}_1,$$

and

$$\mathbb{G}_2 = \{x_k^{(2)} = \phi_2^{-1}(kh) : k = -N, \ldots, N\} \subset \mathbb{X}_2.$$

By applying the Sinc quadrature method to the integral term $\mathcal{Q}_{\mathbb{X}}(V^j)$ in $\mathbb{X}_1$ and $\mathbb{X}_2$ separately, we get

$$\begin{aligned} \mathcal{Q}_{\mathbb{X}}(V^j)(x) &= \int_{\mathbb{X}_1} V^j(y)\varphi(y-x)\mathrm{d}y + \int_{\mathbb{X}_2} V^j(y)\varphi(y-x)\mathrm{d}y \\ &\approx h \sum_{i=-N}^{N} \frac{V^j(x_i^{(1)})\varphi(x_i^{(1)} - x)}{\phi_1'(x_i^{(1)})} + h \sum_{i=-N}^{N} \frac{V^j(x_i^{(2)})\varphi(x_i^{(2)} - x)}{\phi_2'(x_i^{(2)})}, \end{aligned} \tag{5.39}$$

where we also use the fact that the quadrature nodes coincide with the Sinc spatial grids in $\mathbb{X}_1$ and $\mathbb{X}_2$, respectively.

Obviously, we can also evaluate the integrals $\mathcal{Q}_{\mathbb{X}}(V^j)(x)$ at all the Sinc spatial grids simultaneously by one matrix-vector product. Unfortunately, the applicability of FFT is deprived as the associated matrix forfeits the Toeplitz structure due to the non-uniform spatial grids, and therefore the matrix-vector multiplication is very costly. Alternatively, we present a fast algorithm for evaluating the summations (5.39) with a Gaussian kernel $\varphi(x)$ in the next subsection.

5.2 IFGT for approximating the integrals

In this subsection, we employ the IFGT [23, 39] to speed up the evaluation of (5.39). For clarity of exposition, let $n = 2N + 1$ and define a new set of grid points

$$\widehat{y}_i = \begin{cases} x_{i+N+1}^{(1)}, & \text{if} \quad -n \le i \le -1, \\ \xi, & \text{if} \quad i = 0, \\ x_{i-N-1}^{(2)}, & \text{if} \quad 1 \le i \le n. \end{cases}$$

Given any spatial grid point $\bar{x} \in \mathbb{G}_1 \cup \mathbb{G}_2$, recall from (5.39) that

$$\mathcal{Q}_{\mathbb{X}}(V^j)(\bar{x}) = \sum_{i=-n}^{n} \widehat{w}_i \varphi(\widehat{y}_i - \bar{x}),$$

where

$$\widehat{w}_i = \begin{cases} hV^j(\widehat{y}_i)/\phi_1'(\widehat{y}_i), & \text{if} \quad -n \le i \le -1, \\ 0, & \text{if} \quad i = 0, \\ hV^j(\widehat{y}_i)/\phi_2'(\widehat{y}_i), & \text{if} \quad 1 \le i \le n. \end{cases}$$

Substituting the expression of $\varphi(z)$ given in (2.2), we get

$$\mathcal{Q}_{\mathbb{X}}(V^j)(\bar{x}) = \sum_{i=-n}^{n} w_i e^{-(y_i-\bar{x})^2/\tau^2},$$

where $y_i = \widehat{y}_i - \mu, w_i = \widehat{w}_i/(\sqrt{2\pi}\gamma)$ and $\tau = \sqrt{2}\gamma$. Straightforward computation of the above summations at all the Sinc spatial grids takes $\mathcal{O}(n^2)$ operations. We mention here that the set of source points y_i and target points $\bar{x}$ are disparate unless $\mu = 0$.

For any point $y_* \in \mathbb{X}$, the Gauss function at $\bar{x}$ can be written as

$$\begin{aligned} e^{-(y_i-\bar{x})^2/\tau^2} &= e^{-[(y_i-y_*)-(\bar{x}-y_*)]^2/\tau^2} \\ &= e^{-(y_i-y_*)^2/\tau^2} \cdot e^{-(\bar{x}-y_*)^2/\tau^2} \cdot e^{2(y_i-y_*)(\bar{x}-y_*)/\tau^2}. \end{aligned} \tag{5.40}$$

In the last term of equation (5.40), the first exponential $e^{|y_i-y_*|^2/\tau^2}$ and the second exponential $e^{|\bar{x}-y_*|^2/\tau^2}$ depend only on the source points y_i and the target points $\bar{x}$, respectively. However, the source points and target points are involved for the third exponential $e^{2(y_i-y_*)(\bar{x}-y_*)/\tau^2}$. The innovation of IFGT is to separate these involved terms via Taylor's series expansion of the exponentials. For a fixed $\bar{x}$, we have the following series expansion

$$e^{2(y_i-y_*)(\bar{x}-y_*)/\tau^2} = \sum_{k=0}^{p_i-1} \frac{2^k}{k!} \left[\left(\frac{y_i - y_*}{\tau} \right) \left(\frac{\bar{x} - y_*}{\tau} \right) \right]^k + R_{p_i}(y_i),$$

where the truncation number p_i for each source y_i is chosen based on the prescribed error ϵ and its distance from the expansion center, i.e. $|y_i - y_*|$, and the error term $R_{p_i}(y_i)$ satisfies

$$|R_{p_i}| \le \frac{2^{p_i}}{p_i!} \left(\frac{|y_i - y_*|}{\tau} \right)^{p_i} \left(\frac{|\bar{x} - y_*|}{\tau} \right)^{p_i} e^{2|y_i-y_*||\bar{x}-y_*|/\tau^2}.$$

Hence, we further get the expansion

$$e^{-(y_i-\bar{x})^2/\tau^2} = e^{-|y_i-y_*|^2/\tau^2} e^{-|\bar{x}-y_*|^2/\tau^2} \left\{ \sum_{k=0}^{p_i-1} \frac{2^k}{k!} \left[\left(\frac{y_i - y_*}{\tau} \right) \left(\frac{\bar{x} - y_*}{\tau} \right) \right]^k \right\} + E_{ik},$$

where the new error terms E_{ik} satisfy

$$|E_{ik}| \le \frac{2^{p_i}}{p_i!} \left(\frac{|y_i - y_*|}{\tau} \right)^{p_i} \left(\frac{|\bar{x} - y_*|}{\tau} \right)^{p_i} e^{-(|y_i-y_*|-|\bar{x}-y_*|)^2/\tau^2}.$$

Let $p_{\max} = \max\limits_{-n \le i \le n} \{p_i\}$ and $\chi_{\{k \le p_i - 1\}}$ be a character function for $k \le p_i - 1$. Ignoring the error terms E_{ik} and rearranging the terms, we get an approximation

$$\begin{aligned} \widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x}) &= \sum_{i=-n}^{n} w_i e^{-|y_i-y_*|^2/\tau^2} e^{-|\bar{x}-y_*|^2/\tau^2} \left\{ \sum_{k=0}^{p_i-1} \frac{2^k}{k!} \left[\left(\frac{y_i - y_*}{\tau} \right) \left(\frac{\bar{x} - y_*}{\tau} \right) \right]^k \right\} \\ &= \sum_{i=-n}^{n} w_i e^{-|y_i-y_*|^2/\tau^2} e^{-|\bar{x}-y_*|^2/\tau^2} \left\{ \sum_{k=0}^{p_{\max}-1} \frac{2^k}{k!} \left[\left(\frac{y_i - y_*}{\tau} \right) \left(\frac{\bar{x} - y_*}{\tau} \right) \right]^k \chi_{\{k \le p_i - 1\}} \right\} \\ &= \sum_{k=0}^{p_{\max}-1} F_k e^{|\bar{x}-y_*|^2/\tau^2} \left(\frac{\bar{x} - y_*}{\tau} \right)^k, \end{aligned}$$

where

$$F_k = \left[\frac{2^k}{k!} \sum_{i=-n}^{n} w_i e^{-|y_i - y_*|^2/\tau^2} \left(\frac{y_i - y_*}{\tau} \right)^k \chi_{\{k \le p_i - 1\}} \right].$$

The coefficients F_k can be evaluated separately in $\mathcal{O}(n)$ operations. Evaluations of $\widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x})$ at n different target points $\bar{x} \in \mathbb{G}_1 \cup \mathbb{G}_2$ are also of $\mathcal{O}(n)$ complexity. The efficiency of computing $\widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x})$ rather than $\mathcal{Q}_{\mathbb{X}}(V^j)(\bar{x})$ is determined by $p_{\max}$, and the approximating accuracy depends on the truncated error terms E_{ik}. For the prescribed error ϵ, our goal is to find an approximation $\widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x})$ satisfying

$$\frac{\left| \widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x}) - \mathcal{Q}_{\mathbb{X}}(V^j)(\bar{x}) \right|}{\sum_{i=-n}^{n} |w_i|} \le \epsilon.$$

This inequality can be fulfilled by requiring $|E_{ik}| \le \epsilon$.

Because abridging the distance $|y_i - y_*|$ diminishes the bound of error terms E_{ik}, we bestow the clustering algorithm in [12] to separate the sources $\{y_i\}_{-n}^{n}$ into L clusters with center z_l and radius r_l, denoted by $\mathbb{Y}_l$ for $l = 1, 2, \ldots, L$. After that, the center of the Taylor series for y_i is set to be the center of the cluster to which y_i belongs. The clustering steps can be performed in $\mathcal{O}(Ln)$ operations. Moreover, due to the rapid decay of the Gaussian function, the contribution of sources in the lth cluster can be ignored if $|\bar{x} - z_l| > \hat{r}_l \equiv r_l + \tau\sqrt{\log(1/\epsilon)}$.

So now the approximation of $\mathcal{Q}_{\mathbb{X}}(V^j)(\bar{x})$ becomes

$$\widehat{\mathcal{Q}}_{\mathbb{X}}(V^j)(\bar{x}) = \sum_{l=1}^{L} \chi_{\{|\bar{x} - z_l| \le \hat{r}_l\}} \left\{ \sum_{k=0}^{p_{\max}-1} F_k^l e^{-|\bar{x} - z_l|^2/\tau^2} \left(\frac{\bar{x} - z_l}{\tau} \right)^k \right\},$$

where

$$F_k^l = \left[\frac{2^k}{k!} \sum_{y_i \in \mathbb{Y}_l} c_i e^{-|y_i - z_l|^2/\tau^2} \left(\frac{y_i - z_l}{\tau} \right)^k \chi_{\{k \le p_i - 1\}} \right].$$

Taken as a whole, the complexity is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ operations; see [23, 39] for more details. In fact, a callable software package with an automatic tuning procedure for choosing the related parameters, such as L, z_l, r_l, etc., has been developed in [23]. We call this package directly in our numerical experiments.

To verify the linear complexity of above introduced IFGT, we test the problem of computing the summations

$$W_{\mathbb{X}}(x_j) = \sum_{i=-n}^{n} w_i e^{-(y_i - x_j)^2/\tau^2}, \quad j = -n, ..., n, \tag{5.41}$$

with $\{x_j\}_{j=-n}^{n} = \mathbb{G}_1 \cup \mathbb{G}_2 \subset \mathbb{X}$, $\{y_i\}_{i=-n}^{n} = \{x_j - \mu\}_{j=-n}^{n}$ and $\{w_j\}_{j=-n}^{n}$ being uniform random numbers in $(0, 1)$. Here we choose $\mathbb{X} = [-4, 4]$, $\mu = 0$, $\tau = \sqrt{2}/2$ and run for different prescribed error ϵ. Table 1 shows comparisons of the CPU-time expended in the computation by the IFGT algorithm and direct calculation, for different values of n. The "Relative error" is computed by

$$\frac{\|W^{\text{IFGT}} - W^{\text{Direct}}\|_\infty}{\sum_{i=-n}^{n} |w_i|},$$

Table 1: Relative error and CPU time of the IFGT and direct method for (5.41).

	$\epsilon = 10^{-8}$			$\epsilon = 10^{-12}$			
	Relative	IFGT		Relative	IFGT		Direct
n	Error	Time	$(p_{\max}, L)$	Error	Time	$(p_{\max}, L)$	Time
65	5.89e-10	0.0012	(19,9)	2.55e-14	0.0015	(47,5)	0.0018
129	1.07e-10	0.0017	(20,9)	5.12e-14	0.0022	(26,9)	0.0106
257	7.26e-10	0.0032	(20,9)	1.03e-14	0.0038	(26,9)	0.0204
513	1.34e-10	0.0057	(20,9)	1.73e-14	0.0067	(26,9)	0.0697
1025	4.46e-11	0.0094	(20,9)	3.74e-15	0.0122	(27,9)	0.2656
2049	1.03e-09	0.0293	(20,9)	3.88e-14	0.0226	(26,9)	1.0181
4097	3.05e-10	0.0332	(21,9)	6.86e-15	0.0437	(27,9)	3.9135

where W^{IFGT} and W^{Direct} denote the computed vector of summations by the IFGT and direct method, respectively. From Table 1, we see that the CPU time for the IFGT algorithm grows linearly, in contrast to the quadratic growth of that for the direct method. Moreover, the maximum relative absolute error of all the IFGT approximations are less than the prescribed error ϵ and the IFGT parameters $p_{\max}$ and L almost keep the same as the grids size increases.

5.3 Computational costs

We implement the proposed scheme for solving the PIDE (2.1) as the following steps.

1. Discretize the PIDE (2.1) in time by IMEX scheme (2.3) to gain a BVP (2.12) for every time step.

2. Approximating the BVP (2.12) by the Sinc DDM gives the Sinc DDM system (4.34). The complexity of constructing the whole system is of $\mathcal{O}(n)$, observing that only the first and last column of Toeplitz matrices T_2 and T_1 are computed, respectively.

3. In the right-hand side of the Sinc DDM system (4.34), the integral term $\mathcal{Q}(V)$ in (2.13) is approximated by IFGT with $\mathcal{O}(n)$ complexity.

4. The Sinc DDM system (4.34) is then solved by GMRES with preconditioner (4.37). By the analysis in [28], the complexity is roughly of $\mathcal{O}(n \log n)$.

5. Repeat Steps 2 to Steps 4 for m times as required by the IMEX scheme with S stages of extrapolation.

To summarize, our proposed method costs nearly $\mathcal{O}(n \log n)$ operations per time step and converges exponentially, which is very attractive to those applications demanding high accuracy.

6 Numerical results

In this section, we use our algorithm to solve the PIDE for the European call option and the double-barrier call option. All numerical experiments are implemented using MATLAB

7.12 on a laptop with Intel(R) Pentium(R) CPU P6100@2.00GHz and 6GB RAM.

To avoid deteriorating the exponential convergence rate of the spatial discretization, we take a sufficently large stage number S and initial time steps J in the Richardson extrapolation method. The total number of time steps is $m = J(2^S - 1)$. The IFGT prescribed error is set to $\epsilon = 10^{-12}$ for the sake of high accuracy. The right preconditioned GMRES is employed to solve Sinc DDM systems with stopping criterion $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\| \leq tol$, where $\boldsymbol{r}_k$ is the residual vector after k iterations. We take the approximated solution at the last time step as the initial guess. The setup parameters for the Sinc method are given by $\zeta = 1, d = \frac{\pi}{2}$ and $h = \sqrt{\frac{\pi d}{N}}$. In pricing European call options in Merton jump-diffusion model, we use the analytic series solution given by the Merton's formula [22] for computing errors of approximated solutions, while for double-barrier options, the benchmark price is computed with fine enough spatial grids and time steps as suggested in [13].

We use "E_K" to denote the error of the approximate solution at the the strike point $x_K = 0$, "E_∞" the maximum absolute error on the corresponding uniform grid points, "Iter" the average number of iterations in each time step, and "Time" the total computational time. The column "Precond." means that the preconditioner P in (4.37) is applied and "Unprecond. " represents no preconditioner is used.

6.1 European call option.

In the first numerical example, we price European call options defined by the parameters

$$T = 1,\ \sigma = 0.2,\ r = 0,\ \nu = 0,\ \mu = 0,\ \lambda = 0.1,\ \gamma = 0.5,\ \text{and}\ K = 1.$$

The same parameters were used by Almendral and Oosterlee [1], in which the truncated domain was set to be $[x_{\min}, x_{\max}] = [-4, 4]$ and the PIDE was discretized by a second order finite difference method (SOFDM), backward difference formula of second order (BDF2), and composite trapezoidal rule in space, time, and integral, respectively.

As an improvement, Sachs and Strauss [31] proposed to eliminate the convection term by a variable transformation and then solve the resulting symmetric Toeplitz systems by preconditioned conjugate gradient (PCG) method with efficient circulant preconditioners. We report their results (using Strang's preconditioner) in Table 2, where the computational time increases by a factor of about 4 since their approach only needs $\mathcal{O}(n \log n)$ operations per time step. Clearly, their method achieves a second order convergence.

In Table 3, we report the error results and computational time for solving the PIDE using our proposed method, where the Sinc DDM systems are solved by preconditioned GMRES and unpreconditioned GMRES, respectively. Here we set $\delta = 0.04$ on the coarsest grid. The sharp decrease of the average iteration numbers for the preconditioned GMRES versus that of unpreconditioned GMRES confirms the efficiency of the proposed preconditioner (4.37). We observe an exponential convergence rate of the maximum error from the column "E_∞", where we applied Sinc interpolation to get the approximations on the same uniform grids as computed by SOFDM. It's interesting to remark that the approximation at the strike point x_K is much more accurate since more grid points are put around x_K.

In Figure 1, we plot the point-wise error of Sinc approximations $U_{\mathbb{X}_1}(x)$ and $U_{\mathbb{X}_2}(x)$ over all Sinc grid points. The maximum error points appear at the right boundary as reported

Table 2: The errors and CPU time for pricing the European call option by SOFDM with $tol = 10^{-8}$ for PCG (Reference price $V(0,T) = 0.094135507492$).

n	m	$V(0,T)$	E_K	E_∞	Iter	Time
65	5	0.0931631619	9.7e-04	2.4e-03	4	0.05
129	10	0.0941273894	8.1e-06	6.0e-04	4	0.12
257	20	0.0939631865	1.7e-04	1.8e-04	4	0.46
513	40	0.0941045133	3.1e-05	3.8e-05	4	1.85
1025	80	0.0941322670	3.2e-06	9.6e-06	4	7.40
2049	160	0.0941354703	3.7e-08	2.4e-06	4	29.68
4097	320	0.0941348093	7.0e-07	7.3e-07	4	120.93
8193	640	0.0941353746	1.3e-07	1.5e-07	4	573.05

Table 3: The errors and CPU time for pricing the European call option by Sinc DDM with $tol = 10^{-14}$ for GMRES (Reference price $V(0,T) = 0.094135507492$).

		Precond.					Unprecond.	
$2n$	$m(J,S)$	$V(0,T)$	E_K	E_∞	Iter	Time	Iter	Time
66	15(1,4)	0.0938427070	2.9e-04	6.9e-03	14	0.45	58	1.16
130	62(2,5)	0.0941373514	1.8e-06	3.4e-04	13	0.93	94	7.17
258	124(4,5)	0.0941362147	7.1e-07	2.2e-06	11	2.07	159	31.32
514	248(8,5)	0.0941355058	1.7e-09	2.0e-08	8	4.57	233	134.09
1026	496(16,5)	0.0941355072	2.8e-10	2.8e-10	6	10.79	314	742.70

Figure 1: The point-wise error of approximations $V(x,T)$ over Sinc grid points.

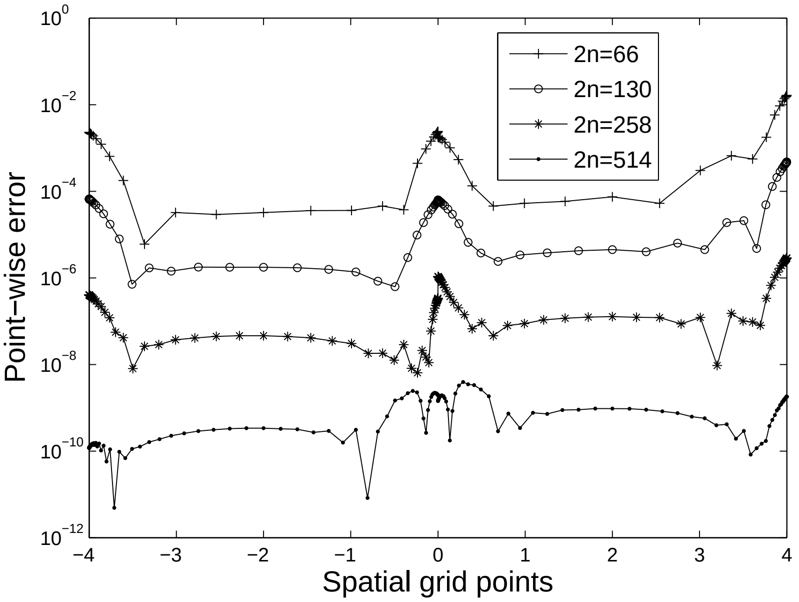


Figure 2: The maximum errors of Sinc DDM and SOFDM with respect to spatial grids.

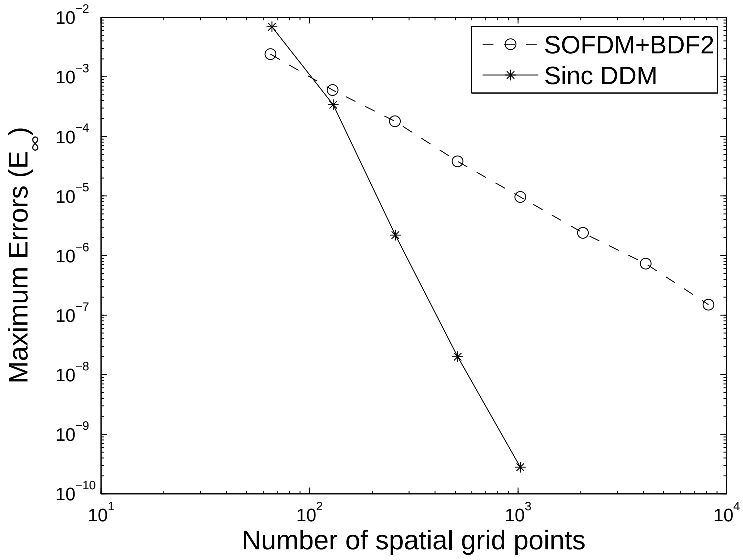
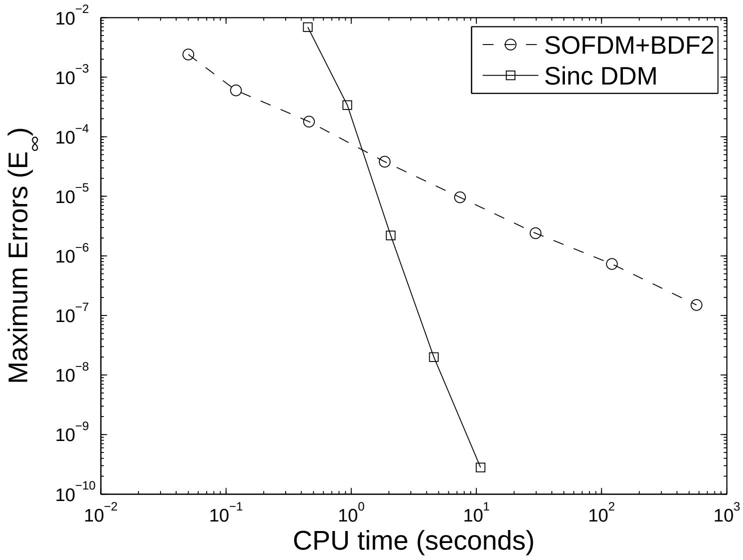


Figure 3: The maximum errors of Sinc DDM and SOFDM with respect to CPU time.



in [1], but this substantial contrast is finally eliminated providing sufficient refined meshes. The central peaks indicate the concentration of grid cells around the strike point is very necessary; Otherwise the error around the strike point will not be reduced so dramaticly.

To highlight the exponential accuracy of our method, we compare the error decays between Sinc DDM and SOFDM in Figure 2. Evidently, our proposed Sinc DDM could produce much more accurate approximations than the SOFDM providing the grid sizes are greater than a hundred. Moreover, we plot the error decays with respect to the computational time in Figure 3, which shows Sinc DDM becomes much faster than SOFDM when the seeking accuracy is higher than 10^{-5}. For instance, it costs Sinc DDM only 2 seconds to attain 10^{-6} maximum error, while SOFDM needs about 30 seconds to achieve the same level of accuracy. Thus, Sinc DDM becomes more advantagous than SOFDM if we are targeting high accuracy.

6.2 Double-barrier call option.

In the second example, we present numerical results for double-barrier call options with the lower barrier $x_d = \log(80/100)$ and the upper barrier $x_u = \log(120/100)$. The model parameters are given as

$$T = 1,\ \sigma = 0.1,\ r = 0.05,\ \nu = 0.02,\ \mu = -0.05,\ \lambda = 3,\ \gamma = 0.086,\ K = 100,$$

which was also experimented by Feng and Linetsky [13]. In Table 4, we report the error results and computational time for solving the PIDE using our method, where the error E_K is computed by differencing with the benchmark value given in the last row. Here we set $\delta = 0.002$ on the coarsest grid. These results again illustrate the exponential convergence rate of Sinc DDM and the efficiency of the suggested preconditioner, even though high order numerical methods for pricing double-barrier options are generally deemed to be more difficult. Our benchmark price (1.964728493) at $x_K = 0$ have about four more correct significant digits than the one (1.96473) given in [13]. We also plot the approximated option prices $V(x, T)$ on the Sinc grid points in Figure 4, which shows more grid points are concentrated at the strike price $x_K = 0$ and boundaries.

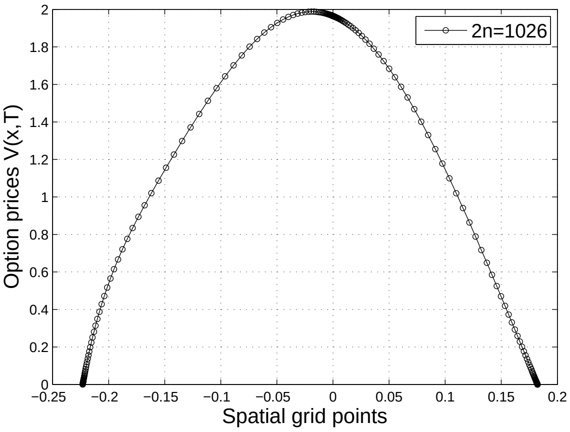
Table 4: The errors and CPU time for pricing the double-barrier call option by Sinc DDM with *tol* $= 10^{-14}$ for GMRES.

		Precond.				Unprecond.	
$2n$	m(J,S)	$V(0,T)$	E_K	Iter	Time	Iter	Time
34	56(8,3)	1.964102816	6.3e-04	19	1.4	36	3.1
66	112(16,3)	1.964548338	1.8e-04	19	3.6	68	7.4
130	480(32,4)	1.964777230	4.9e-05	17	15.4	127	89.7
258	960(64,4)	1.964730400	1.9e-06	16	39.0	214	417.3
514	1920(128,4)	1.964728498	5.4e-09	14	55.2	313	2011.6
1026	7936(256,5)	1.964728493	—	9	189.9	—	—

7 Conclusions

In this contribution, we have considered an exponentially accurate Sinc DDM discretization in space with an extrapolated IMEX-Euler scheme in time for pricing both European and double-barrier options under Merton's jump-diffusion model. The difficulty stems from the non-smooth payoff functions is remedied by combining the non-overlapping domain decomposition method with Sinc method. The special jump sizes distribution of Merton's jump-diffusion model allows us to take advantage of the IFGT when calculating the integral summations. With the proposed preconditioner, we solved the resulting Toplitz-like dense linear systems very effectively by preconditioned GMRES.

Figure 4: The approximated $V(x,T)$ of the double-barrier call option on Sinc grid points.



It is worthwhile to remark that our method does not apply to American option problem, which is more computationally challenging due to discontinuities in their second derivatives at the critical stock prices [21]. It usually requires adaptive time stepping for high-order schemes to realize a high-order convergence [13]. We focus on developing high-order scheme in space, while high-order convergence in time are fulfilled by utilizing the Richardson extrapolation technique. More effective time schemes need to be developed to reduce the computational cost. As the first try of applying Sinc method to option pricing, only the classic Merton jump-diffusion model was considered and other modern jump-diffusion models including Kou's model, the variance gamma model, and the CGMY model will be investigated in future work.

Acknowledgments.

The authors would like to thanks Spike T. Lee for plenty of insightful discussions and constructive suggestions on polishing this paper. The authors are also grateful to the anonymous referees for their useful comments and suggestions.

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] K. Amin, Jump diffusion option valuation in discrete time, Journal of Finance, 48 (1993), pp. 1833–1863.

[3] L. Andersen and J. Andreasen, Jump-diffusion processes: volatility smile fitting and numerical methods for pricing, Rev. Deriv. Res., 4 (2000), pp. 231–262.

[4] F. Black and M. Scholes, The pricing of options and corporate liabilities, J. Polit. Economy, 81 (1973), pp. 637–654.

[5] M. Broadie and J. Detemple, Option pricing: valuation models and applications, Manage. Sci., 50 (2004), pp. 1145–1177.

[6] M. Broadie and Y. Yamamoto, Application of the fast Gauss transform to option pricing, Manage. Sci., 49 (2003), pp. 1071–1088.

[7] R. Chan and X. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, PA, 2007.

[8] R. Cont and P. Tankov, Financial Modelling with Jump Processes, Chapman &Hall/CRC, Boca Raton, FL, 2004.

[9] R. Cont and E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM J. Numer. Anal., 43 (2005), pp. 1596–1626.

[10] C. Dawson, Q. Du, and T. Dupont, A finite difference domain decomposition algorithm for numerical solution of the heat equation, Math. Comp., 57 (1991), pp. 63–71.

[11] 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.

[12] T. Feder and D. Greene, Optimal algorithms for approximate clustering, in STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing, 1988, pp. 434–444.

[13] L. Feng and V. Linetsky, Pricing options in jump-diffusion models: an extrapolation approach, Oper. Res., 56 (2008), pp. 304–325.

[14] L. Greengard and J. Strain, The fast Gauss transform, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 79–94.

[15] R. Horn and C. Johnson, Matrix Analysis, Cambridge University Press, 1990.

[16] S. Lee and H. Sun, Fourth order compact boundary value method for option pricing with jumps, Advances in Applied Mathematics and Mechanics, 1(2009), pp. 845–861.

[17] R. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM, Philadelphia, PA, 2007.

[18] J. Lund and K. Bowers, Sinc Methods for Quadrature and Differential Equations, SIAM, Philadelphia, PA, 1992.

[19] N. Lybeck and K. Bowers, Domain decomposition in conjunction with sinc methods for Poisson's equation, Numer. Methods Partial Differential Equations, 12 (1996), pp. 461–487.

[20] N. Lybeck and K. Bowers, Sinc methods for domain decomposition, Appl. Math. Comput., 75 (1996), pp. 13–41.

[21] A. Mayo, High-order accurate implicit finite difference method for evaluating american options, The European Journal of Finance, 10(2004), pp. 212–237.

[22] R. Merton, Option pricing when underlying stock returns are discontinuous, J. Financial Eco., 3 (1976), pp. 125–144.

[23] V. Morariu, B. Srinivasan, V. Raykar, R. Duraiswami and L. Davis, Automatic online tuning for fast Gaussian summation, in Advances in Neural Information Processing Systems (NIPS), 2008.

[24] A. Morlet, N. Lybeck and K. Bowers, The Schwarz alternating sinc domain decomposition method, Appl. Numer. Math., 25 (1997), pp. 461–483.

[25] A. Morlet, N. Lybeck and K. Bowers, Convergence of the sinc overlapping domain decomposition method, Appl. Math. Comput., 98(1999), pp. 209–227.

[26] M. Ng, Fast iterative methods for symmetric sinc-Galerkin systems, IMA J. Numer. Anal., 19 (1999), pp. 357–373.

[27] M. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, 2004.

[28] M. Ng and D. Potts, Fast iterative methods for sinc systems, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 581–598.

[29] A. Quarteroni and A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, 1999.

[30] Y. Saad. Iterative Methods for Sparse Linear Systems, Second Edition, SIAM, Philadelphia, PA, 2003.

[31] E. Sachs and A. Strauss, Efficient solution of a partial integro-differential equation in finance, Appl. Numer. Math. , 58(2008), pp. 1687–1703.

[32] F. Stenger, Numerical Methods Based on Sinc and Analytic Functions, Springer-Verlag, New York, 1993.

[33] J. Stoer and R. Bulirsch, Introduction to numerical analysis, Springer-Verlag, New York, 1993.

[34] M. Sugihara and T. Matsuo, Recent developments of the Sinc numerical methods, J. Comput. Appl. Math., 164/165 (2004), pp. 673–689.

[35] D. Tangman, A. Gopaul and M. Bhuruth, Exponential time integration and Chebychev discretization schemes for fast pricing of options, Appl. Numer. Math., 58 (2008), pp. 1309–1319.

[36] 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.

[37] D. Tavella and C. Randall, Pricing Financial Instruments: the Finite Difference Method, John Wiley & Sons, Chichester, UK, 2000.

[38] J. Toivanen, Numerical valuation of European and American options under Kou's jump-diffusion model, SIAM J. Sci. Comput., 30 (2008), pp. 1949–1970.

[39] C. Yang, R. Duraiswami, N.A. Gumerov and L. Davis, Improved Fast Gauss Transform and Efficient Kernel Density Estimation, in ICCV, 2003, pp. 464–471.

[40] J. Zhao, W. Dai, and T. Niu, Fourth-order compact schemes of a heat conduction problem with Neumann boundary conditions, Numer. Methods Partial Differential Equations, 23 (2007), pp. 949–959.