Computational Economics
https://doi.org/10.1007/s10614-022-10339-2

Method of Lines for Valuation and Sensitivities of Bermudan Options

Purba Banerjee[1] · **Vasudeva Murthy**[2] · **Shashi Jain**[3]

Accepted: 15 October 2022

Abstract

In this paper, we present a computationally efficient technique based on the *Method of Lines* for the approximation of the Bermudan option values via the associated partial differential equations. The method of lines converts the Black Scholes partial differential equation to a system of ordinary differential equations. The solution of the system of ordinary differential equations so obtained only requires spatial discretization and avoids discretization in time. Additionally, the exact solution of the ordinary differential equations can be obtained efficiently using the exponential matrix operation, making the method computationally attractive and straightforward to implement. An essential advantage of the proposed approach is that the associated Greeks can be computed with minimal additional computations. We illustrate, through numerical experiments, the efficacy of the proposed method in pricing and computation of the sensitivities for a European call, cash-or-nothing, powered option, and Bermudan put option.

Keywords Method of lines · Bermudan options · Fast Greeks · Finite difference method for option pricing

Vasudeva Murthy and Shashi Jain these authors contributed equally to this work.

✉ Shashi Jain
shashijain@iisc.ac.in

Purba Banerjee
purbab@iisc.ac.in

Vasudeva Murthy
vasu@tifrbng.res.in

[1] Department of Mathematics, Indian Institute of Science, Bangalore 560012, India

[2] TIFR Centre For Applicable Mathematics, Bangalore 560065, India

[3] Department of Management Studies, Indian Institute of Science, Bangalore 560012, India

Published online: 17 November 2022

1 Introduction

Pricing an option with early exercise features, such as Bermudan options, is a problem of practical importance with numerous pricing methods proposed in the literature, each with its own set of advantages. When fast and accurate evaluation of the option price is the objective, the Fourier-based methods are commonly used numerical techniques. Some of the proposed Fourier-based pricing methods include the Fourier–Cosine method (Fang & Oosterlee, 2009),data-driven Fourier–Cosine method (Leitao Rodriguez et al., 2017), and fast Fourier transform-based approach (Carr & Madan, 1999). Monte Carlo based schemes are popular for pricing the early exercise options under multidimensional stochastic processes. The traditional lattice-based methods are often impractical in such cases due to the curse of dimensionality. Popular simulation-based pricing models include Longstaff and Schwartz method (Longstaff & Schwartz, 2001), the stochastic mesh method by Broadie and Glasserman (2004), and the stochastic grid bundling method Jain and Oosterlee (2015). There has been an increasing interest in neural network-based pricing methods for early exercise options. Some of the recent ones include Andersson and Oosterlee (2021), Lokeshwar et al. (2021), and Becker et al. (2019).

For low dimensional options, often finite difference methods are used for approximating the solution to the underlying partial differential equation, as approximate option prices for a grid of underlying values (see for instance De Graaf et al., 2014) can then be obtained. Solving efficiently the partial differential equations (PDEs) for American styled options using finite difference schemes have been extensively studied in Brennan and Schwartz (1977), In't Hout and Foulon (2010), and Haentjens and in't Hout (2015). The ability to compute prices along a spatial and temporal grid of underlying stochastic states is helpful for the simulation of future exposure. Future exposure is required, for instance, for credit valuation adjustment or for the risk management purposes, such as determining the potential future exposure of a particular position.

In this paper we propose an accurate and efficient approach based on the Method of Lines (MOL) to compute the value of Bermudan and European options for a grid of underlying values at any time prior to maturity. The MOL for solving evolutionary partial differential equation (PDEs) consists of two parts, first involves discretisation of the space variable and then writing the PDE as a system of ordinary differential equations (ODEs). The second part involves efficiently solving the system of ODEs so obtained. The main advantage of the proposed approach is that it solves the ODEs so obtained exactly, by writing the solution to the ODE as an exponential of a matrix. As most modern numerical packages come with efficient solvers for computing the exponential of matrices, we can efficiently compute the exact solution for the ODE. The above is possible in the Black–Scholes framework as the coefficients of the PDE are time-independent and the PDE is linear.

Several researchers have worked on the MOL in the past few decades, most notable among which include Hamdi et al. (2007), Schiesser and Griffiths (2009),

and Lee and Schiesser (2003) who provide a general elucidation to the method. For a detailed discussion on the application of time discrete MOL for pricing of options, the readers can refer (Meyer, 2014). Similar time-discretised approaches have been implemented for pricing of American put option in the Black–Scholes framework (Meyer & Van der Hoek, 1997), for put options under jump-diffusion dynamics (Meyer, 1998), call options under stochastic volatility (Adolfsson et al., 2013), American options with stochastic volatility and interest rates (Kang & Meyer, 2014), American options under a regime-switching GBM (Chiarella et al., 2016). The method proposed here is closely related to the approach proposed in Horng et al. (2019), and Meyer and Van der Hoek (1997). Compared to these methods stated above, our approach does not require time discretisation, and the option value can be found as a continuous function of time.

Some of the recent regulatory requirements, such as the exchange of initial margin computed based on the ISDA standard initial margin model (SIMM), require efficient computation of the sensitivities of the derivative with respect to its underlying risk factors. Further, to compute the associated margin valuation adjustment (MVA), to manage the funding costs for posting the initial margin over the lifetime of a derivative, sensitivities along the paths are required (see Jain et al., 2019). Computation of the sensitivities along the paths poses noteworthy computational challenges for all the underlying risk factors.

Some of the key features of the study presented in this paper are:

- We provide the exact solution for the ODEs obtained using MOL from the Black Scholes PDEs for European options.
- The scheme avoids discretisation in the time domain, and the solution is closed form in time. Therefore, one can obtain the approximate option price at any time instant before the maturity of the option.
- The matrix expressions for the Greeks: delta, theta, gamma, and vega are obtained, which allows evaluating the various sensitivities at the spatial grid points with a few additional computations. This feature makes the approach attractive when sensitivities along scenarios are required.
- We extend the approach to Bermudan options by formulating a Bermudan option as a sequence of European options with appropriate pay-offs.

The paper is organised as follows: We first begin with the formulation of the problem and defining the notations used in Sect. 2. Section 3 describes in detail the methodology used. In Sect. 4, the expressions for the Greeks are obtained. Section 5 presents detailed numerical examples for European call, cash or nothing, powered option, and Bermudan options to illustrate the efficiency of the proposed method. Finally, we provide some conclusions in Sect. 6.

2 Problem Formulation

A Bermudan option is defined as an option where the buyer has the right to exercise at a set of discrete time points. We denote the exercise times by

$$\mathcal{T} \equiv \{t_1, \ldots, t_e, \ldots, t_E = T\},$$

where $0 = t_0 \leq t_1 \leq t_2 \ldots t_{E-1} \leq T$, T is the maturity date of the option, and E denotes the number of early exercise opportunities. For ease of notation, we assume that the exercise dates are equally spaced, i.e., $t_{e+1} - t_e = \Delta t$.

The pay-off received by the holder of the option upon exercising his rights at the opportunity $t_e, e \in \{1, \ldots, E\}$ is given by $\phi(S_{t_e})$. The continuation value, or the value of the option if the holder decides not to exercise and continue holding the option at t_e, is defined as:

$$c(t_e, S_{t_e}) = e^{-r\Delta t}\mathbb{E}\left[U\left(t_{e+1}, S_{t_{e+1}}\right) \mid \mathcal{F}_{t_e}\right],$$

where r is the risk free rate (we assume it to be constant), $\mathcal{F}_{t_e}$ is the filtration associated with the stochastic process S_t, the expectation is taken under the risk-neutral measure, and $U\left(t_{e+1}, S_{t_{e+1}}\right)$, is the option value function at t_{e+1}.

The option value at any time t is then solved using the following dynamic programming formulation. The value of the option at the terminal time T is given by,

$$U\left(t_E, S_{t_E}\right) = \max\left(\phi(S_{t_E}), 0\right).$$

Recursively, moving backwards in time, the following iteration is then solved. Given $U\left(t_{e+1}, S_{t_{e+1}}\right)$, has already been resolved,

$$\begin{aligned} c(t_e, S_{t_e}) &= e^{-r\Delta t}\mathbb{E}\left[U\left(t_{e+1}, S_{t_{e+1}}\right) \mid \mathcal{F}_{t_e}\right] \\ U\left(t_e, S_{t_e}\right) &= \max\left(c(t_e, S_{t_e}), \phi(S_{t_e})\right). \end{aligned} \tag{1}$$

The value of the option at t_0, assuming there is no exercise opportunity at t_0 is then given by $U(t_0, S_{t_0}) = c(t_0, S_{t_0})$

3 Methodology

Under the Black–Scholes framework (Black & Scholes, 1973), when the underlying stock follows the geometric Brownian motion (GBM), the price of the European option, U, satisfies the following PDE,

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU = 0, \tag{2}$$

where the underlying S follows the GBM process,

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where r is the risk-free rate, μ is the drift of the underlying process, σ is the implied volatility, and W_t is the standard Brownian motion.

In order to numerically solve (2) for $t \in [0, T]$, we first transform it into the familiar forward in time parabolic partial differential equation by replacing U with u, S with x and t with $\tau = T - t$, thereby obtaining the following initial-boundary value problem for $u(\tau, x)$,

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u}{\partial x^2} + rx\frac{\partial u}{\partial x} - ru, \quad x > 0, \quad \tau \in (0, T], \tag{3}$$

With the initial condition given by Eq. (4).

$$u(0, x) \equiv u_0(x) = \phi(x). \tag{4}$$

The value of Bermudan option, with early exercise dates $\mathcal{T} \equiv \{t_1, \ldots, t_e, \ldots, t_E = T\}$, can be formulated as follows. Define $\tau_e = T - t_{E-e}$, for $e = 0, 1, \ldots, E$, such that $\tau_0 = 0$ and $\tau_E = T$. The fair value function u of Bermudan option satisfies the PDE (3) with natural boundary condition on each time interval $\left[\tau_{e-1}, \tau_e\right)$, for $e = 1, 2, \ldots, E$. The initial condition for τ_0 is

$$u(\tau_0, x) = \phi(x),$$

while the initial condition for solving the PDE (3) in the interval (τ_{e-1}, τ_e), $e = 2, \ldots, E$ is given by

$$u(\tau_{e-1}, x) = \max\left(\phi(x), \lim_{t \uparrow \tau_{e-1}} u(t, x)\right) \tag{5}$$

Condition (5) is non-linear and arises from the early exercise feature of Bermudan options and represents the optimal exercise condition. We solve the PDE (3) in the interval $[\tau_{e-1}, \tau_e)$, to obtain the initial condition for the subsequent interval, i.e. $[\tau_e, \tau_{e+1})$.

Since numerically we cannot work with the infinite spatial domain $[0, \infty)$, the spatial domain is truncated at $x = L$. We use three types of grids in the spatial dimension:

1. The first one being where, a family of uniform grid $\eta_n = \frac{n}{N}$, $n \leq N$, (N is the number of discretization points) are defined in the interval [0, 1], which generates a two parameter family of quasi-uniform grid

$$x_n = c\frac{\eta_n}{d - \eta_n}, \tag{6}$$

   where $\eta_n \in [0, 1], x_n \in [0, \infty)$. This map has two control parameters $c > 0$, and $d > 1$ (see Fazio & Jannelli, 2014). Once a uniform mesh has been generated in [0, 1], we shall use the above mentioned transformation to generate a non-uniform grid, called NUG1 in our text, with $x_0 = 0$, and truncate x_{N+1} at L.
2. For the second non-uniform grid, termed NUG2, we generate x_n's using the map,

$$x_n = K \times e^{y_n}, \tag{7}$$

where K is the strike of the option, and y_n's are uniformly distributed over the interval $[-m_1, m_2]$, for choice of $m_1, m_2 > 0$.

3. The third one is constructed as a uniform grid x_n called UG1, in the interval $[S_{\min}, S_{\max}]$, where $0 < S_{\min} < S_0 < S_{\max}$.

Following this, we set $x_i - x_{i-1} = h_i$, for $i = 1....N$ and $h_{N+1} = L - x_N$. We then reduce (3) to a set of ODEs by approximating the spatial derivatives in the following manner (see Volders, 2014): at each grid point x_i, we approximate the option value by

$$u(\tau, x_i) \approx U_i(\tau).$$

Further, the approximation of the first derivative is given by,

$$u_x(\tau, x_i) \approx \frac{U_{i+1}(\tau) - U_{i-1}(\tau)}{h_i + h_{i+1}} = DU_i(\tau). \tag{8}$$

Analogously, the approximation for the second derivative is

$$u_{xx}(\tau, x_i) \approx \frac{2U_{i-1}(\tau)}{h_i(h_i + h_{i+1})} - \frac{2U_i(\tau)}{h_i h_{i+1}} + \frac{2U_{i+1}(\tau)}{h_{i+1}(h_i + h_{i+1})} = D^2 U_i(\tau). \tag{9}$$

It should be noted that for a uniform mesh we have $h_i = h$, hence the above approximations would in the case of uniform mesh coincide with the familiar second order approximations to the first and second derivatives, i.e.,

$$u_x(\tau, x_i) \approx \frac{U_{i+1}(\tau) - U_{i-1}(\tau)}{2h}, \quad u_{xx}(\tau, x_i) \approx \frac{U_{i+1}(\tau) - 2U_i(\tau) + U_{i-1}(\tau)}{h^2}.$$

Upon replacing the spatial derivatives in (3) by the approximations stated above, we obtain the following system of ODEs,

$$\frac{dU_i(\tau)}{d\tau} = \frac{1}{2}\sigma^2 {x_i}^2 D^2 U_i(\tau) + r x_i D U_i(\tau) - r U_i(\tau), \quad \text{for} \quad i = 1, \ldots N. \tag{10}$$

In order to obtain the value of the option, the matrices corresponding to the coefficients of the derivative approximations need to be defined. Consequently, the matrix corresponding to the second order derivatives is defined as follows,

$$A = \begin{bmatrix} \gamma_1 & \alpha_1 & \ldots & 0 \\ \alpha_{-2} & \gamma_2 & \ldots & 0 \\ \ldots\ldots\ldots & & & \\ 0 & \ldots\ldots\ldots & \alpha_{-N} & \gamma_N \end{bmatrix}$$

where,

$$\gamma_j = -\frac{\sigma^2 {x_j}^2}{h_j h_{j+1}}, \quad j = 1, \ldots N,$$
$$\alpha_j = \frac{\sigma^2 {x_j}^2}{(h_j + h_{j+1}) h_{j+1}}, \quad j = 1, \ldots N$$
$$\alpha_{-j} = \frac{\sigma^2 {x_j}^2}{(h_j + h_{j+1}) h_j} \quad j = 1, \ldots N$$

and the matrix corresponding to the first order derivatives is

$$B = \begin{bmatrix} 0 & b_1 & .... & 0 \\ -b_2 & ..... & & \\ ..... & .... & ... & b_{N-1} \\ 0 & .... & -b_N & 0 \end{bmatrix}$$

where,

$$b_j = \frac{r x_j}{h_j + h_{j+1}}, \quad j = 1, \ldots N.$$

.

Finally the zero-th order term in (3) is incorporated in the following matrix,

$$C = \begin{bmatrix} -r & 0 & ... & 0 \\ 0 & .... & ... & \\ .. & .... & ... & .. \\ 0... & .. & 0 & -r \end{bmatrix}$$

with the boundary conditions being given by,

$$F(\tau) = [(\alpha_{-1} - b_1) u(\tau, 0), 0, \ldots 0, (\alpha_N + b_N) u(\tau, L)]^T. \tag{11}$$

Using these, the ODE (10) can be compactly written as,

$$\frac{dU}{d\tau} = AU + BU + CU + F(\tau). \tag{12}$$

For the interval $[\tau_{e-1}, \tau_e)$, the initial condition would be,

$$U(\tau_{e-1}) = [u(\tau_{e-1}, x_1), u(\tau_{e-1}, x_2), \ldots .u(\tau_{e-1}, x_N)]^\top.$$

3.1 Exact solution to the ODEs

Denoting $\zeta = A + B + C$, the ODE (12) can be readily solved to obtain the following solution,

$$U(\tau) = e^{\zeta(\tau-\tau_{e-1})}U(\tau_{e-1}) + \int_{\tau_{e-1}}^{\tau} e^{\zeta(\tau-s)}F(s)ds, \tag{13}$$

where $\tau \in [\tau_{e-1}, \tau_e)$. The above is possible based on the observation that ζ is not a function of time. The computation of the expression above requires the explicit definition of $F(s)$ from Eq. (11). While $u(\tau, 0)$ is in general known to be equal to $u(\tau_{e-1}, 0)$, $u(\tau, L)$ is often not known. Hence, we draw inspiration from Kangro and Nicolaides (2000) and Higham (2004) and the fact that for large enough L it is known that the standard European put option will be far out of money, while a standard call option will be far in money, at all times, from (1) we can approximate the underlying option's value as $e^{-r(\tau-\tau_{e-1})}u(\tau_{e-1}, L) \approx u(\tau_{e-1}, L)$, which readily yields,

$$F(\tau) = F = \begin{bmatrix} (\alpha_{-1} - b_1)u(\tau_{e-1}, 0) \\ 0 \\ . \\ . \\ . \\ 0 \\ (\alpha_N + b_N)u(\tau_{e-1}, L) \end{bmatrix}. \tag{14}$$

The approximation $e^{-r(\tau-\tau_{e-1})}u(\tau_{e-1}, L) \approx u(\tau_{e-1}, L)$ is beneficial as the resultant F is a constant, and the second term in (13) can then be integrated exactly to obtain the final expression,

$$U(\tau) = e^{\zeta(\tau-\tau_{e-1})}U(\tau_{e-1}) + \zeta^{-1}(e^{\zeta(\tau-\tau_{e-1})} - 1)F. \tag{15}$$

If we do not make the assumption of constant F, one could use the exponential *Runge–Kutta* schemes discussed in Hochbruck and Ostermann (2010). To compute Eq. (15), one needs to calculate the exponential of a matrix, which has an efficient implementation in most modern numerical libraries (see Appendix 7.1 for details on matrix exponential).

We can obtain an exact solution for any τ in the interval $[\tau_{e-1}, \tau_e)$ for all the spatial grid points which makes this approach attractive for exposure computation, as proposed in De Graaf et al. (2014), where finite difference methods can be combined with Monte Carlo based methods. In the section to follow, we extend the above approach to obtain the various sensitivities, which again can be combined with Monte Carlo methods to obtain sensitivities along scenarios (see, for instance de Graaf et al., 2017).

3.2 Parameter and Computational Considerations

In order to determine the value of parameters: $c, m_1, m_2, S_{\min}$ and $S_{\max}$, used for generating the three grids, we apply a bisection method on a coarser grid in each of the cases. There could be a more thorough heuristic approach to arrive at the parameters, which we leave as a part of future research.

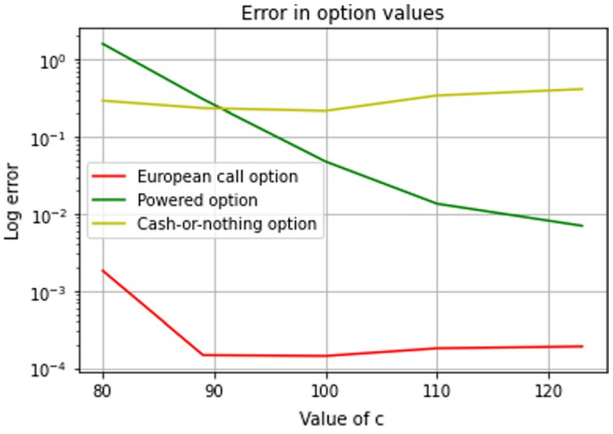

Fig. 1 Error at $S_0 = 100$ for European call, cash-or-nothing, and powered options when time to maturity is 1 year, with varying values for parameter c

Figure 1 depicts the changes in the log errors of the European call option, powered option, and cash-or-nothing option for different values of the parameter c, while keeping the rest of the parameters constant, and the number of grid points fixed at $N = 1000$.

For the European call option, we see that the error reduces as we increase the value of c and beyond $c \approx 90$, the errors are fairly stable. For the Powered option increasing c progressively reduces the error, although beyond $c = 123$, there is a slight rise in the error. For cash-or-nothing, the error is stable until $c < 100$, beyond which there seems to be a slight increase in the errors. Based on these heuristics, we pre-select the value of c with the lowest error for each corresponding experiment.

Arriving at the matrices A, B, and C in Eq. (13) for N grid points involves $\mathcal{O}(N)$, computations. The computationally intensive steps involve matrix inversion, which involves $\mathcal{O}(N^3)$ operations, and computing the exponential of the matrix, which again has $\mathcal{O}(N^3)$ operations. Therefore, overall the computational complexity of the method is $\mathcal{O}(N^3)$.

3.3 Study of Convergence and Stability

In this section we discuss about the convergence and stability of the MOL method. While we do our error analysis for the uniform grid, the results obtained are similar for the non-uniform grid as well. We start our analysis with the equation under consideration,

$$\frac{dU_i(\tau)}{d\tau} = \frac{1}{2}\sigma^2 {x_i}^2 D^2 U_i(\tau) + r x_i D U_i(\tau) - r U_i(\tau), \quad \text{for} \quad i = 1, \ldots N. \tag{16}$$

Following the analysis as done in Appendix 7.3, we get,

$$D^2U(\tau) = \frac{U(\tau, x+h) - 2U(\tau, x) + U(\tau, x-h)}{h^2} = U_{xx}(\tau, x) + \frac{h^2}{12}U_{xxxx}(\tau, x) + \mathcal{O}(h^3). \tag{17}$$

In an analogous manner, using Taylor's expansion, we get,

$$DU(\tau, x) = \frac{U(\tau, x+h) - U(\tau, x-h)}{2h} = U_x(\tau, x) + \frac{h^2}{3!}U_{xxx}(\tau, x) + \mathcal{O}(h^3) \tag{18}$$

Now, if $U(\tau, x)$ denotes the exact solution of the system (16), then, as shown in Appendix 7.3, the local accuracy is given by

$$R = -\sigma^2 x^2 \frac{h^2}{24}U_{xxxx}(\tau, x) + \mathcal{O}(h^3) - \frac{rx}{3!}h^2 U_{xxx}(\tau, x) + \mathcal{O}(h^3) \tag{19}$$

Equation (19) shows that the local accuracy of the MOL method behaves as $\mathcal{O}(h^2)$.

It must also be noted from the above expression that the convergence of our method only depends upon the spatial discretization and we are exact with respect to time.

To address the issue of stability, it must be noted that the numerical scheme we solve is given by the following ODE,

$$\frac{dU}{d\tau} = \zeta U + F, \tag{20}$$

whose solution is given by (15). As this is a linear ODE, following (Strang, 2006), it can be concluded that this system will be stable when the real part of the Eigen values of ζ are negative. Therefore, in order to check the stability, one should first check the Eigen values of ζ. In all the numerical examples we considered, we found that the corresponding ζ had negative Eigen values. The reader can refer to Appendix 7.3 for a brief discussion about stability.

This is one of the inherent advantages of our method that the stability of the system of ODEs is driven only by the underlying constant matrix ζ and is not influenced by an additional effect from any time discretisation in the model. For a detailed stability analysis of solving such a special system of equations through a method of exponential integrators, the reader can refer to Brachet et al. (2020) and Buvoli and Minion (2022).

4 Greeks Using MOL

The Greeks of an option represent the sensitivity of the option's price with respect to changes in underlying parameters involved in the definition of the option. The primary Greeks of interest include: delta $\Delta = \frac{\partial U}{\partial S}$, vega $\nu = \frac{\partial U}{\partial \sigma}$, theta $\Theta = \frac{\partial U}{\partial t}$, and rho $\rho = \frac{\partial U}{\partial r}$. Amongst second-order Greeks, gamma, $\Gamma = \frac{\partial^2 U}{\partial S^2}$. is often computed. Based

on the methodology described above, we present simple extensions to compute the Greeks, not just at the initial point but for any t at the generated spatial grid.

As a direct consequence of the approximation (8), one obtains the matrix corresponding to the first order and second order derivatives as

$$\Delta(\tau) = D_1 U(\tau) + D_2 \tag{21}$$

and the value of Gamma(Γ) is given by the matrix product,

$$\Gamma(\tau) = D_1^2 U(\tau) + D_1 D_2 + D_2, \tag{22}$$

where the matrix D_1 and D_2 are defined as:

$$D_1 = \begin{bmatrix} 0 & d_1 & .... & 0 \\ -d_2 & ..... & & \\ ..... & .... & ... & d_{N-1} \\ 0 & .... & -d_N & 0 \end{bmatrix}$$

with,

$$d_j = \frac{1}{h_j + h_{j+1}}, \quad j = 1, \ldots N.$$

and,

$$D_2 = \begin{bmatrix} -d_1 \times u(\tau, 0) \\ 0 \\ . \\ . \\ 0 \\ d_N \times u(\tau, L) \end{bmatrix}$$

and $U(\tau)$ is obtained using Eq. (15).

In order to calculate Theta(Θ) one can differentiate $U(\tau)$ in (15) to obtain the following formula for Θ,

$$\frac{\partial U(\tau)}{\partial t} = -\zeta e^{\tau\zeta} U_0 - e^{\tau\zeta} F \tag{23}$$

For the calculation of Vega(ν), a direct application of calculus theory to (15) and the identity $\zeta.\zeta^{-1} = I_N$ yields the partial differential of $U(\tau)$ with respect to σ,

$$\frac{\partial U(\tau)}{\partial \sigma} = \tau e^{\tau\zeta} A' U_0 - \zeta^{-1} A' \zeta^{-1} (e^{\tau\zeta} - I) F + \tau \zeta^{-1} e^{\tau\zeta} A' F + \zeta^{-1} (e^{\tau\zeta} - I) F' \tag{24}$$

where A' and F' are the matrices obtained by term-by-term differentiation of the matrices A and F (14) respectively with respect to σ.

Analogously one obtains the formula for Rho(ρ),

$$\frac{\partial U(\tau)}{\partial r} = \tau e^{\tau\zeta}(B' + C')U_0 - \zeta^{-1}(B' + C')\zeta^{-1}(e^{\tau\zeta} - I)F + \tau\zeta^{-1}e^{\tau\zeta}(B' + C')F + \zeta^{-1}(e^{\tau\zeta} - I)F'_r \tag{25}$$

where B', C' and F'_r are the matrices obtained by term-by-term differentiation of the matrices B, C and F respectively with respect to r. The corresponding derivations of the formulae obtained for the Θ, ν and ρ can be found in the Appendix 7.2. The sensitivities are usually computed only at S_0, but here we also obtain the sensitivities at underlying prices corresponding to the other grid points for any $t \in [0, T]$, without significant additional computations.

Efficient computation of sensitivities of derivatives with respect to the underlying risk factors is becoming increasingly important, not only from the perspective of hedging and managing risk but also to meet certain regulatory requirements. ISDA SIMM[1] requires the sensitivities of the derivatives for computing the initial margin to be exchanged for over the counter derivatives portfolio. Managing the funding risk associated with the exchange of the initial margin, through the life of the derivative position, involves computing the sensitivities along the simulated scenarios. The proposed method gives a closed-form expression of the various sensitivities as a function of time for a grid of underlying values. Following the approach discussed in de Graaf et al. (2017), forward sensitivities of options can be simulated by generating the underlying Monte Carlo scenarios and interpolating the sensitivities obtained at the fixed grid points in the proposed method.

5 Numerical Examples

To illustrate the performance and efficiency of the method, we start with the assumption that the underlying asset price process S_t follows the GBM process and then apply the MOL to compute the value of certain types of European options, a basic European call option, a powered option, and finally a cash-or-nothing option; all of which are special cases of the standard Bermudan option defined earlier, where the number of exercise opportunities, E, is set to one. We then report results for a Bermudan put option and numerically study its convergence when an increasing number of spatial grid points are used.[2] The following parameters are used throughout for the European styled options: $\sigma = 0.3$, $r = 0.03$, initial stock price $S_0 = 100$, strike price $K = 100$ and $T = 1$.

Our first step is to generate a uniform mesh $\{\eta_n = n \times dx\}_{n=0}^{N}$ in [0, 1] with $dx = 1/N$ and then append an additional point $\eta_{N+1} = 1.1$. A non-uniform mesh, NUG1: $\{x_n\}_{n=0}^{N+1}$, is generated using the transformation stated in (6), with the value $d = 1.2$ and appropriate values of c, as stated in the respective tables. A brief

[1] INTERNATIONAL SWAPS AND DERIVATIVES ASSOCIATION: ISDA SIMM TM,1 Methodology, Version 2.0 (December, 2017). https://www.isda.org/a/oFiDE/isda-simm-v2.pdf.

[2] The Python code for all the experiments reported here is available at https://github.com/PurbaBanerjee.

Table 1 Value of the European call option obtained by the MOL and the corresponding log errors

N	u(13.28330)	Absolute error	Log error	CPU time (sec)
100	13.25	$2e^{-2}$	− 3.7	0.03
200	13.27	$5e^{-3}$	− 5.3	0.10
400	13.282	$1e^{-3}$	− 6.7	0.41
800	13.2830	$3e^{-4}$	− 8.1	2.30
1600	13.2832	$6.3e^{-5}$	− 9.7	12.61

We take $c = 110$, the parameter for generating the first grid NUG1 through Eq. 6

discussion on choice of the parameter c is provided in Sect. 3.2. We insert in the appropriate position, the point $x = 100$, in case of the European options, and $x = 40$, in case of the Bermudan put option, and evaluate the accuracy of the method at these points.

Similarly, the grids NUG2 and UG3 are generated for an appropriate choice of the parameters: m_1, m_2, $S_{\min}$ and $S_{\max}$, respectively for each of European styled options.

5.1 European Call Option

As the first numerical test we consider a European call option, whose payoff is given by $\phi(x) = (x - K)^+$. For the European call option, the closed-form solution of the Black–Scholes equation is given by,

$$u(\tau, x) = xN(d_1) - Ke^{-r\tau}N(d_2),$$
$$d_1 = (ln(x/K) + (r + 0.5\sigma^2)\tau)/(\sigma\sqrt{\tau}), \quad d_2 = d_1 - \sigma\sqrt{\tau}$$

where, $N(d) = (1/\sqrt{2\pi})\int_{-\infty}^{d} e^{-0.5x^2}dx$, is the cumulative distribution function for the standard normal distribution. The initial conditions for solving this option using the MOL are:

$$u(0, x) = \phi(x), \tag{26}$$

and it also satisfies,

$$u(\tau, x) \approx \phi(x) \quad \text{as} \quad x \to \infty \tag{27}$$

Table 1 reports the values and errors of the European call option obtained by the method on using the first non-uniform grid NUG1, along with the computational(CPU) times, when an increasing number of grid points, N, are used. The exact value of the option is 13.28330.

As a comparison, in Jeong et al. (2018) in order to get an accuracy of $1.73e^{-4}$, 16717 points were used for time discretization, 17141 points for spatial discretization and a CPU time of 19.25 seconds was incurred.

Another interesting point to be noted is that, while in Jeong et al. (2018), we cannot then obtain the value of the option at another arbitrary time point between

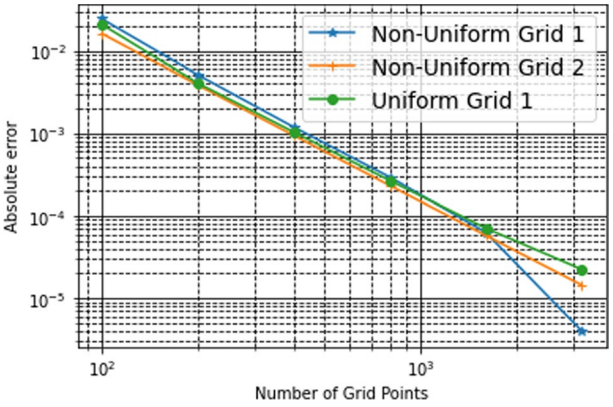

Fig. 2 Error in the price of the European call option at the initial point $S_0 = 100$, when time to maturity is 1 year. The grid parameters being: $c = 110, m_1 = -1.5, m_2 = 1.5, S_{\min} = 30, S_{\max} = 350$

Table 2 Delta and Gamma of the European call option and their corresponding log errors taking $c = 110$ in NUG1

N	Δ(0.59870632)	Log error (Δ)	Γ(0.01288894)	Log error (Γ)
100	0.6	− 3.3	0.0127	− 8.4
200	0.6	− 4.2	0.0127	− 9.2
400	0.6	− 5.4	0.01285	− 10.4
800	0.6	− 5.4	0.01285	− 10.4
1600	0.6	− 6.2	0.01287	− 11.3

0 and T, with our method, such additional computations can be done in merely a few seconds. Also, the computational time for the other grids, NUG2 and UG3 are similar.

Figure 2 illustrates the convergence of the method for an increasing number of grid points in case of each of the three grids. It was observed that the numerical convergence of the method in this case is almost 2, which is along the expected lines.

The Greeks of the European call option are given by Haug (1997),

$$\Delta = N(d_1), \quad \Gamma = \frac{N'(d_1)}{\sigma x\sqrt{\tau}}, \quad \Theta = -\frac{\sigma x N'(d_1)}{2\sqrt{\tau}} - rKe^{-r\tau}N(d_2)$$
$$\nu = x\sqrt{\tau}N'(d_1), \quad \rho = \tau x e^{-r\tau}N(d_2).$$

Table 2 reports the Delta and Gamma values for the European call option obtained by the MOL along with their respective errors in log scale. We see that fairly accurate values are obtained with relatively few grid points.

Table 3 Vega, rho, and theta of the European call option and their corresponding log errors taking $c = 110$ in NUG1

n	ν(38.667)	Log error	ρ(46.5873)	Log error	Θ (−7.19764)	Log error
100	38.7	− 2.4	46.6	− 4.0	− 7.2	− 4.3
200	38.68	− 4.1	46.59	− 4.9	− 7.2	− 5.9
400	38.67	− 5.6	46.589	− 6.1	− 7.198	− 7.4
800	38.667	− 7.0	46.5878	− 7.5	− 7.1977	− 8.8
1600	38.667	− 8.4	46.5874	− 8.8	− 7.19767	− 10.2

Table 4 Value of the Powered option obtained by the MOL and it's corresponding log errors taking $c = 123$ for generating NUG1

N	u(676.758)	Absolute error	Log error	CPU time (sec)
100	677.4	$6e^{-1}$	− 0.4	0.01
200	677.4	$2e^{-1}$	− 1.8	0.06
400	676.8	$4e^{-2}$	− 3.1	0.35
800	676.76	$9e^{-3}$	− 4.6	2.12
1600	676.76	$3e^{-3}$	− 5.7	16.63

Table 3 reports the vega, rho, and theta values for the European call option. Again we see that fairly accurate values are obtained with relatively few grid points.

5.2 Powered Option

Next , following Jeong et al. (2018) we consider a powered option whose pay-off function at maturity T is given by $\phi(x) = \max(x - K, 0)^p$, where p is a constant (called the power). The initial conditions of the PDE (3) for a powered option are given by Eq. (5), similar to the European call option.

The closed-form solution of the powered option is

$$u(\tau, x) = \sum_{q=0}^{p} \frac{p!}{q!(p-q)!} x^{p-q}(-K)^q e^{(p-q-1)(r+0.5(p-q)\sigma^2)\tau} N(d_{p,q}),$$

where, $d_{p,q} = [ln(x/K) + (r + (p - q - 0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$. In the example we choose a value of $p = 2$.

Table 4 reports the values of the powered option obtained by the implementation of the MOL for different values of grid points, N and their corresponding log errors and CPU times for NUG1. The exact value of the option given in Jeong et al. (2018) is 676.758.

Figure 3 illustrates the convergence of the powered option with increasing number of grid points. A convergence of order almost 2 was observed numerically in this case as well.

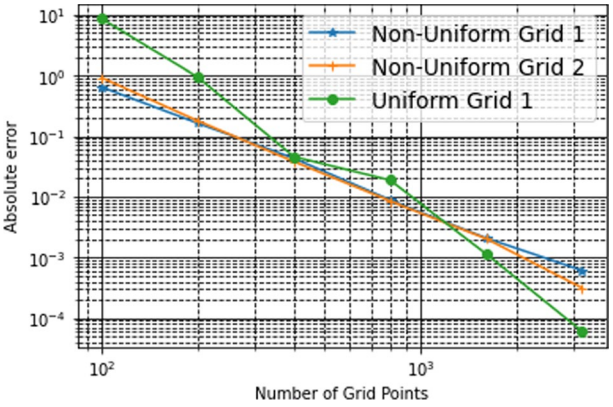

Fig. 3 Error in the price of the European powered option at the initial point $S_0 = 100$, when time to maturity is 1 year. The respective grid parameters are: $c = 123, m_1 = -1.6, m_2 = 1.8, S_{\min} = 30, S_{\max} = 890$.

Table 5 Delta and Gamma of the Powered option and their corresponding log errors taking $c = 123$ in NUG1

N	Δ(40.102)	Log error (Δ)	Γ(1.598)	Log error (Γ)
100	41.2	0.1	1.592	− 5.1
200	41.1	0.1	1.592	− 5.1
400	41.1	0.02	1.592	− 5.1
800	40.4	− 1.1	1.596	− 6.3
1600	40.107	− 5.3	1.598	− 7.7

Table 6 Vega, Rho and Theta of the powered option and their corresponding log errors taking $c = 123$ in NUG1

N	ν(4795.291)	Log error (ν)	ρ(3333.420)	Log error (ρ)	Θ(−819.296)	Log error (Θ)
100	4795.4	− 2.4	3347.2	2.6	− 819.7	− 0.8
200	4795.3	− 3.9	3336.8	1.2	− 819.4	− 2.2
400	4795.293	− 6.3	3334.3	− 0.2	− 819.3	− 3.6
800	4795.295	− 5.4	3333.6	− 1.5	− 819.3	− 4.9
1600	4795.292	− 6.4	3333.47	− 2.9	− 819.298	− 6.2

In contrast, for the same case, in Jeong et al. (2018) you get an accuracy of $2.54e^{-2}$ where 4183 points are used for time discretization, 4395 points were used for spatial discretization and a CPU time of 1.24 s.

As mentioned earlier, the calculation of the option values and their Greeks can be done at any time point, unlike in Jeong et al. (2018) and the additional time required is negligible.

Table 5 reports the values and the respective errors of the delta and gamma values in log scale, obtained by the MOL, where the reference values (from Jeong et al., 2018) are reported in parenthesis.

Table 7 Value of the cash-or-nothing option obtained by the MOL and it's corresponding log errors taking $c = 89$ in NUG1

N	u(46.587)	Absolute error	Log error	CPU time (sec)
100	43.2	3.3	1.2	0.02
200	45.5	1.1	0.1	0.02
400	46.0	$6e^{-1}$	− 0.6	0.22
800	46.3	$3e^{-1}$	− 1.2	1.43
1600	46.4	$2e^{-1}$	− 1.9	8.84

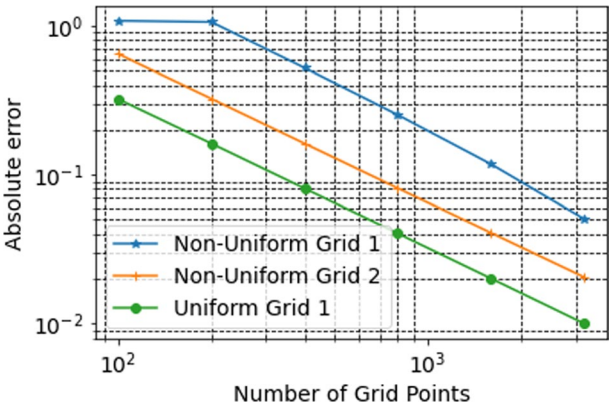

Fig. 4 Error in the price of the Cash-or-nothing at the initial point $S_0 = 100$, when time to maturity is 1 year. The values of the grid parameters are as follows: $c = 89, m_1 = -1, m_2 = 1, S_{\min} = 50, S_{\max} = 300$

We also report the vega, rho, and theta values of the powered option in Table 6.

5.3 Cash-or-Nothing Option

The cash-or-nothing option, with maturity T, pays at maturity, an amount C, provided the value of the underlying asset is greater than K, and no pay-off otherwise. For the experiment, we set $C = K = 100$. The exact solution of the cash-or-nothing option is given by:

$$u(\tau, x) = Ce^{-r\tau}N(d_2).$$

Table 7 reports the values of the cash-or-nothing option obtained for different number of grid points, N and the corresponding log errors for NUG1, with exact value of the option given in Jeong et al. (2018) being 46.587.

In Jeong et al. (2018) in order to obtain an accuracy of $6.93e^{-4}$, a time discretization of 1050 grid points and a spatial discretization of 1156 points were used, with a CPU time of 0.1 s.

Figure 4 illustrates the convergence of the method for an increasing number of grid points for each of the three grids. It was observed numerically that the

Table 8 Delta and Gamma of the cash-or-nothing option and their corresponding log errors taking $c = 89$ in NUG1

N	$\Delta(1.289)$	Log error (Δ)	$\Gamma(-0.011)$	Log error (Γ)
100	1.27	− 3.9	− 0.007	− 5.5
200	1.289	− 5.9	− 0.010	− 6.5
400	1.288	− 6.6	− 0.010	− 7.0
800	1.288	− 7.0	− 0.010	− 7.4
1600	1.288	− 7.4	− 0.011	− 7.7

Table 9 Vega, Rho and Theta of the cash-or-nothing option and their corresponding log errors taking $c = 89$ in NUG1

N	$\nu(-32.222)$	Log error (ν)	$\rho(82.302)$	Log error (ρ)	$\Theta(2.364)$	Log error (Θ)
100	− 20.7	2.4	84.4	0.7	0.5	0.6
200	− 28.5	1.3	83.2	− 0.2	1.7	− 0.5
400	− 30.3	0.6	82.7	− 0.8	2.0	− 1.2
800	− 31.2	− 0.03	82.5	− 1.4	2.2	− 1.9
1600	− 31.7	− 0.7	82.4	− 2.1	2.2	− 2.5

order of convergence of the method for this option was almost 1. In this case, our method is not as accurate as Jeong et al. (2018) but we have an added advantage of the option value and greeks being calculated at any time point, in almost negligible additional time.

The Greeks of the cash-or-nothing option are given by,

$$\Delta = \frac{Ce^{-r\tau}N'(d_2)}{\sigma x\sqrt{\tau}}, \quad \Gamma = -\frac{Cd_1e^{-r\tau}N'(d_2)}{(\sigma x)^2\tau}$$

$$\nu = -Ce^{-r\tau}\frac{d_1}{\sigma}N'(d_2), \quad \rho = Ce^{-r\tau}\left(-\tau N(d_2) + \frac{\sqrt{\tau}}{\sigma}N'(d_2)\right)$$

$$\Theta = Ce^{-r\tau}\left(rN(d_2)(-\tau N(d_2) + \left(\frac{d_1}{2\tau} - \frac{r}{\sigma\sqrt{\tau}}\right)N'(d_2)\right).$$

The following Tables 8 and 9 report the values of the Greeks obtained by the MOL along with their respective errors in log scale. The exact values of the respective quantities are given in the brackets (Jeong et al., 2018).

5.4 Bermudan Put Option

Table 10 depicts the values, the corresponding log errors and the computational times incurred while implementing the MOL for the Bermudan put option. The reference value for the Bermudan put option is obtained by applying a Fourier–cosine method, as done in the COS method in Fang and Oosterlee (2009).The COS method can, in general, handle more general dynamics for the underlying,

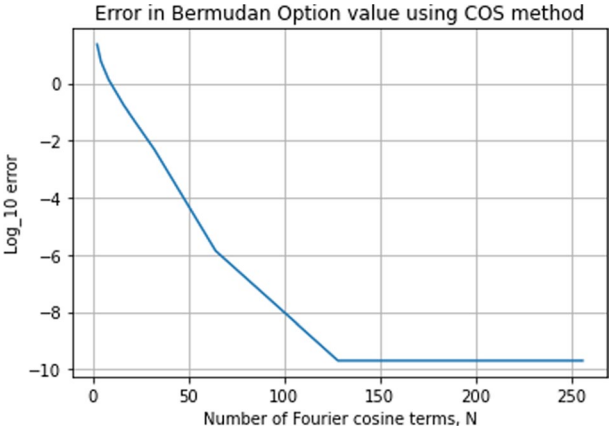

Fig. 5 Error versus number of Fourier cosine terms, N, for a Bermudan put option, using COS method. The parameter values for the Bermudan option are $S_0 = 100, K = 110, T = 1, r = 0.1, \sigma = 0.2, E = 10$

Table 10 Value of the Bermudan put option obtained by the MOL and it's corresponding log errors along with their computational time

n	u(6.04590214)	Log error	CPU time (sec)
250	6.0454	− 7.6	0.24
500	6.0456	− 8.1	1.70
1000	6.0458	− 9.8	10.83
1500	6.0458	− 10.8	27.49
2000	6.0458	− 11.7	67.54
2500	6.0458	− 12.6	144.86

The parameter for NUG1 is taken as $c = 80$.

when compared to the existing methods like Gauss transform (Broadie & Yamamoto, 2003) and the double exponential transformation (Mori & Sugihara, 2001; Yamamoto, 2005) and can price a vector of strike prices simultaneously. Further, the accuracy and the error analysis of this method, as reported in Fang and Oosterlee (2009) compare well with other Fourier transform based methods such as the Carr and Madan (1999) and the CONV (Lord et al., 2008) method. For a detailed error analysis of the COS method implementation in case of European options, the reader can refer to Section 4 of Fang and Oosterlee (2009). We first reproduce in Fig. 5 the convergence results reported in Fang (2010) using our implementation of the COS method. The reference value of 10.4795201 is obtained through finite difference scheme, using 2000 spatial, and 4000 temporal discretization points.

For implementation of the MOL the following parameters were used for the asset price model and the option $\sigma = 0.3, r = 0.06, T = 1$, initial stock price $S_0 = 40$, and the strike $K = 44$. The option has $E = 10$, equally spaced early exercise opportunities. Figure 6 shows the numerical convergence of the method with

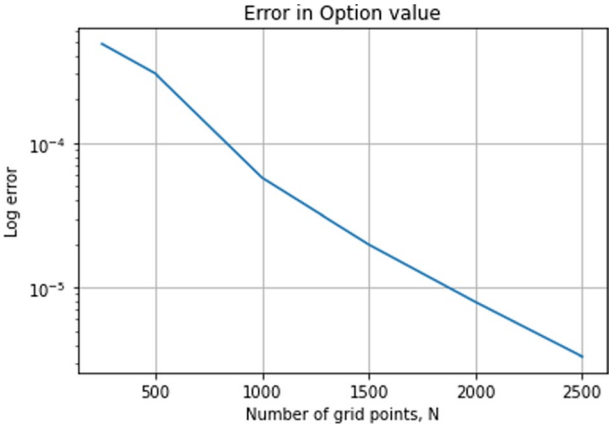

Fig. 6 Error in the price of the Bermudan put option at the initial point $S_0 = 40$, a strike of 44, when time to maturity is 1 year with $E = 10$ equally spaced exercise opportunities. The value of parameter c for generating NUG1 is taken as 80

an increasing number of spatial discretization points. Table 10 reports the computational effort required to achieve different level of accuracy.

With the above numerical experiments, we study the convergence of the MOL with spatial discretization for the various payoffs for the European options and a Bermudan put option. For the non-uniform and uniform spatial grid, we observe a second-order convergence for all the payoffs considered, except for the cash-or-nothing option, where we observe the first-order convergence. The convergence is studied only for spatial discretization as the method is exact in time. We numerically compute for each case that the option Greeks with little additional computational cost. Finally, the numerical experiments demonstrate the efficiency of the method. For instance, it takes 0.24 s to achieve a numerical log error of −7.6 with the MOL.

6 Conclusion

We have presented an approach based on the MOL for obtaining the value of European and Bermudan options and their sensitivities to the various risk factors and model parameters. The MOL approach allows the underlying PDEs to be converted to a system of ODEs, which can be solved using a large choice of available efficient solvers. In the Black Scholes framework, we show that an exact solution to the ODE can be obtained using an exponential of a matrix. This makes the presented approach attractive as then we avoid discretisation in time, which makes the method highly efficient.

For various risk management and regulatory calculations, for instance, simulation of future exposure, one has to compute the value of the derivatives along simulated scenarios at different time points. Following the approach discussed in De Graaf et al. (2014), an efficient solution to the PDE can be combined with Monte Carlo

simulations to efficiently compute the exposure along scenarios using interpolation. With the presented approach, one can obtain the exposure of European and Bermudan options for a grid of underlying states at arbitrary time points. Additionally, the sensitivities at these spatial grid points can also be computed for any time grid.

The numerical results obtained for the European options and the Bermudan option, along with their respective log errors, prove the desired efficiency of the approach in option valuation. We see that fairly accurate solutions can be obtained with relatively few spatial grid points.

We can summarise the benefits and the limitations of our method as follows:

- The first benefit of our method is that we can frame the problem as a system of ODEs which are can be solved to obtain exact solutions.This allows easy evaluation of the option values at current time t_0, and any arbitrary time point before the expiry. This is difficult with most schemes requiring time discretisation, as one would have to interpolate to obtain the option value at an arbitrary time point.
- Our method involves spatial discretization alone and no time discretization. Therefore, the accuracy is exact with respect to time and is only driven by the spatial grid size, as discussed earlier.
- The third important benefit is that the option values as well the Greeks can be calculated directly at any time point, at negligible time expense.
- The stability of our method is only driven by the underlying matrix ζ and hence, is far simple to evaluate.
- The method is attractive when one is concerned with memory requirements, as now only spatial discretisation needs to be stored.
- Further, the method is shown to be applicable to options with early exercise feature.
- As far as the limitations are concerned, the method derived is applicable when volatility is assumed to be constant, as in the Black–Scholes framework, and hence, the framework for stochastic or even local volatility would be fairly different and remains to be a motivation for future exploration for us.
- Further, other mappings could utilise this method, which still needs to be explored.

Appendix

Matrix Exponential

The matrix exponential is defined for $A \in \mathbb{C}^{n\times n}$ by,

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots \cdot \tag{28}$$

We shall now state an important theorem which yields the convergence of the matrix Taylor series (28).

Theorem 7.1 (Convergence of matrix Taylor series) *Suppose f has a Taylor series expansion*,

$$f(z) = \sum_{k=0}^{\infty} a_k (z-\alpha)^k, \quad \left(a_k = \frac{f^{(k)}(\alpha)}{k!} \right) \tag{29}$$

with radius of convergence r. If $A \in \mathbb{C}^{n\times n}$ *then f*(*A*) *is defined and is given by*

$$f(A) = \sum_{k=0}^{\infty} a_k (A-\alpha I)^k \tag{30}$$

if and only if each of the distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ *of A satisfies one of the conditions*

1. $|\lambda_i - \alpha| < r$,
2. $|\lambda_i - \alpha| = r$,

and the series for $f^{(n_i-1)}(\lambda)$ (*where* n_i *is the index of* λ_i) *is convergent at the point* $\lambda = \lambda_i, i = 1 : s$.

Now, if we consider the power series (28), then the corresponding power series is given by,

$$f(z) = \sum_{k=0}^{\infty} \frac{z^k}{k!} \tag{31}$$

whose radius of convergence is,

$$r = \frac{1}{\lim_{n\to\infty} \frac{a_{n+1}}{a_n}}$$

where, $a_n = \frac{1}{n!}$ (since the limit in the denominator exists in this case). Hence, this readily yields

$$r = \frac{1}{\lim_{n\to\infty} \frac{n!}{(n+1)!}} = \infty$$

Therefore, on applying the theorem above we know that the series (28) is defined and further by standard results in analysis (as in Gillespie, 1955), we can differentiate the series term by term to obtain $\frac{d}{dt} e^{At} = Ae^{At} = e^{At}A$.

Another representation of a matrix exponential is,

$$e^A = \lim_{s\to\infty} (I + A/s)^s \tag{32}$$

The formula above is the limit of the first-order Taylor expansion of A/s raised to the power of $s \in \mathbb{Z}$. In a more general setting, we can take the limit as $r \to \infty$ or $s \to \infty$ of r terms of the Taylor expansion of A/s raised to the power of s, thereby generalising both (28) and (32). Results show that this general formula also yields e^A and provides an error bound for finite r and s. We shall not delve into these details but refer the reader to Gillespie (1955) for a detailed analysis of the same.

Derivations of the Greeks

Theta (Θ): Since the matrices do not involve the parameter t, a direct differentiation of the function $U(\tau)$ in (15) with respect to t yields the following formula for Θ,

$$\begin{aligned}\frac{\partial U(\tau)}{\partial t} &= -\zeta e^{\tau\zeta} U_0 - \zeta^{-1}\zeta e^{\tau\zeta} F \\ &= -\zeta e^{\tau\zeta} U_0 - e^{\tau\zeta} F\end{aligned}$$

Vega (ν): The calculation of this is a bit involved since the coefficients of the matrices A and F involve the parameter σ. We first use the identity $\zeta\zeta^{-1} = I$ and differentiate it partially with respect to σ and utilise the chain rule to obtain,

$$\begin{aligned}&\frac{\partial \zeta}{\partial \sigma}\zeta^{-1} + \zeta\frac{\partial \zeta^{-1}}{\partial \sigma} = 0 \\ &\implies \zeta\frac{\partial \zeta^{-1}}{\partial \sigma} = -\frac{\partial \zeta}{\partial \sigma}\zeta^{-1} \\ &\implies \frac{\partial \zeta^{-1}}{\partial \sigma} = -\zeta^{-1}\frac{\partial \zeta}{\partial \sigma}\zeta^{-1}\end{aligned}$$

Now, a simple application of chain rule to (15) , combined with the formula obtained and the straight-forward result $\frac{\partial \zeta}{\partial \sigma} = A'$ yields the formula (24) for ν.

Rho (ρ): An almost identical formulation as that of ν yields the following formula,

$$\frac{\partial \zeta^{-1}}{\partial r} = -\zeta^{-1}\frac{\partial \zeta}{\partial r}\zeta^{-1}$$

Using this and the result $\frac{\partial \zeta}{\partial r} = B' + C'$, one readily obtains the formula (25) for ρ.

Accuracy and Stability of MOL

It can be seen using a direct application of Taylor's expansion that $U(\tau, x)$ satisfies the following equation,

$$U(\tau, x-h) = U(\tau,x) - hU_x(\tau,x) + \frac{h^2}{2}U_{xx}(\tau,x) - \frac{h^3}{3!}U_{xxx}(\tau,x_i) + \frac{h^4}{4!}U_{xxxx}(\tau,x) + \mathcal{O}(h^5)$$

which gives,

$$\frac{U(\tau,x-h)}{h^2} = \frac{U(\tau,x)}{h^2} - \frac{U_x(t,x)}{h} + \frac{U_{xx}(\tau,x)}{2} - \frac{h}{3!}U_{xxx}(\tau,x) + \frac{h^2}{4!}U_{xxxx}(\tau,x) + \mathcal{O}(h^3).$$

Similarly,

$$U(\tau,x+h) = U(\tau,x) + hU_x(t,x) + \frac{h^2}{2}U_{xx}(\tau,x) + \frac{h^3}{3!}U_{xxx}(\tau,x) + \frac{h^4}{4!}U_{xxxx}(\tau,x) + \mathcal{O}(h^5)$$

which leads to

$$\frac{U(\tau,x+h)}{h^2} = \frac{U(\tau,x)}{h^2} + \frac{U_x(t,x)}{h} + \frac{U_{xx}(\tau,x)}{2} + \frac{h}{3!}U_{xxx}(\tau,x) + \frac{h^2}{4!}U_{xxxx}(\tau,x) + \mathcal{O}(h^3).$$

Upon adding these two equations, one obtains the expression for $D^2U(\tau)$ in (17).

Using these and the fact that $U(\tau,x)$ denotes the exact solution of (16), one obtains the local accuracy of the method as follows (Higham, 2004),

$$R = \frac{\partial U(\tau,x)}{\partial \tau} - \frac{1}{2}\sigma^2x^2D^2U(\tau,x) - rxDU(\tau,x) + rU(\tau,x) \tag{33}$$

which, on application of (17) and (18) becomes,

$$R = \frac{\partial U(\tau,x)}{\partial \tau} - \frac{1}{2}\sigma^2x^2U_{xx}(\tau,x) - \sigma^2x^2\frac{h^2}{24}U_{xxxx}(\tau,x) + \mathcal{O}(h^3) - rxU_x(\tau,x) - \frac{rx}{3!}h^2U_{xxx}(\tau,x) + \mathcal{O}(h^3) + rU(\tau,x)$$

Since, $U(\tau,x)$ is a solution to the Black–Scholes equation (3), the above expression simplifies to,

$$R = -\sigma^2x^2\frac{h^2}{24}U_{xxxx}(\tau,x) + \mathcal{O}(h^3) - \frac{rx}{3!}h^2U_{xxx}(\tau,x) + \mathcal{O}(h^3)$$

which gives the desired local accuracy of the method as $\mathcal{O}(h^2)$.

Recall that our system of differential equations described in Eq. (20) has a solution given by (15). This implies that the system is stable if $e^{t\zeta}$ does not blow up. To

ensure this, it is enough if the real part of each of the Eigen values of the associated matrix ζ are negative, since F is a constant matrix in our case (Strang, 2006).

For general cases, though, F is not necessarily constant and is itself a function of time. In such a case, one could use the scheme discussed in Hochbruck and Ostermann (2010).

Acknowledgements This paper is dedicated to the memory of Professor Vasudeva Murthy, who was instrumental in making this work possible.

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

Adolfsson, T., Chiarella, C., Ziogas, A., & Ziveyi, J. (2013). Representation and numerical approximation of American option prices under Heston stochastic volatility dynamics. Technical report

Andersson, K., & Oosterlee, C. W. (2021). A deep learning approach for computations of exposure profiles for high-dimensional Bermudan options. *Applied Mathematics and Computation, 408,* 126332.

Becker, S., Cheridito, P., & Jentzen, A. (2019). Deep optimal stopping. *Journal of Machine Learning Research, 20,* 74.

Black, F., & Scholes, M. (1973). The valuation of options and corporate liabilities. *Journal of Political Economy, 81*(3), 637–654.

Brachet, M., Debreu, L., & Eldred, C. (2020). Comparison of exponential integrators and traditional time integration schemes for the Shallow water equations.

Brennan, M. J., & Schwartz, E. S. (1977). The valuation of American put options. *The Journal of Finance, 32*(2), 449–462.

Broadie, M., & Glasserman, P. (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance, 7,* 35–72.

Broadie, M., & Yamamoto, Y. (2003). Application of the fast Gauss transform to option pricing. *Management Science, 49*(8), 1071–1088.

Buvoli, T., & Minion, M. L. (2022). On the stability of exponential integrators for non-diffusive equations. *Journal of Computational and Applied Mathematics, 409,* 114126.

Carr, P., & Madan, D. (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance, 2*(4), 61–73.

Chiarella, C., Sklibosios Nikitopoulos, C., Schlögl, E., & Yang, H. (2016). Pricing American options under regime switching using method of lines. Available at SSRN 2731087

De Graaf, C. S., Feng, Q., Kandhai, D., & Oosterlee, C. W. (2014). Efficient computation of exposure profiles for counterparty credit risk. *International Journal of Theoretical and Applied Finance, 17*(04), 1450024.

de Graaf, C., Kandhai, D., & Sloot, P. (2017). Efficient estimation of sensitivities for counterparty credit risk with the finite difference Monte Carlo method. *Journal of Computational Finance, 21*(1), 83–113.

Fang, F. (2010). The COS method: An efficient Fourier method for pricing financial derivatives.

Fang, F., & Oosterlee, C. W. (2009). A novel pricing method for European options based on Fourier–cosine series expansions. *SIAM Journal on Scientific Computing, 31*(2), 826–848.

Fazio, R., & Jannelli, A. (2014). Finite difference schemes on quasi-uniform grids for BVPs on infinite intervals. *Journal of Computational and Applied Mathematics, 269,* 14–23.

Gillespie, R. (1955). Principles of Mathematical Analysis. by Walter Rudin. pp. x, 227. 40s. 1953. (Mcgraw-hill)-theory of functions of real variable. by Henry P. Thielman. pp. xiv, 209. 35s. 1953. (Butterworth scientific publications, London). *The Mathematical Gazette*, *39*(329), 258–259

Haentjens, T., & in't Hout, K. J. (2015). ADI schemes for pricing American options under the Heston model. *Applied Mathematical Finance*, *22*(3), 207–237.
Hamdi, S., Schiesser, W. E., & Griffiths, G. W. (2007). Method of lines. *Scholarpedia, 2*(7), 2859.
Haug, E. G. (1997). *The Complete Guide to Option Pricing Formulas*. McGraw-Hill.
Higham, D. J. (2004). *An introduction to financial option valuation: Mathematics, stochastics and computation*.
Hochbruck, M., & Ostermann, A. (2010). Exponential integrators. *Acta Numerica, 19,* 209–286.
Horng, M.-S., Horng, T.-L., & Tien, C.-Y. (2019). A method-of-lines approach for solving American option problems. *Taiwanese Journal of Mathematics, 23*(5), 1253–1270.
In't Hout, K., & Foulon, S. (2010). ADI finite difference schemes for option pricing in the Heston model with correlation. *International Journal of Numerical Analysis & Modeling*, *7*(2)
Jain, S., Leitao, Á., & Oosterlee, C. W. (2019). Rolling Adjoints: Fast Greeks along Monte Carlo scenarios for early-exercise options. *Journal of Computational Science, 33,* 95–112.
Jain, S., & Oosterlee, C. W. (2015). The stochastic grid bundling method: Efficient pricing of Bermudan options and their Greeks. *Applied Mathematics and Computation, 269,* 412–431.
Jeong, D., Yoo, M., & Kim, J. (2018). Finite difference method for the Black–Scholes equation without boundary conditions. *Computational Economics, 51*(4), 961–972.
Kang, B., & Meyer, G. H. (2014). *Pricing an American call under stochastic volatility and interest rates* (pp. 291–314).
Kangro, R., & Nicolaides, R. (2000). Far field boundary conditions for Black–Scholes equations. *SIAM Journal on Numerical Analysis, 38*(4), 1357–1368.
Lee, H. J., & Schiesser, W. E. (2003). *Ordinary and partial differential equation routines in C, C++, Fortran, Java, Maple, and Matlab*.
Leitao Rodriguez, A., Oosterlee, C. W., Ortiz-Gracia, L., & Bohte, S. (2017). On the data-driven COS method. *Applied Mathematics and Computation, 317,* 68–84.
Lokeshwar, V., Bhardawaj, V., & Jain, S. (2021). Neural network for pricing and universal static hedging of contingent claims. *Applied Mathematics and Computation*.
Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies, 14*(1), 113–147.
Lord, R., Fang, F., Bervoets, F., & Oosterlee, C. W. (2008). A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing, 30*(4), 1678–1705.
Meyer, G. H. (2014). *Time-discrete method of lines for options and bonds, the: A PDE approach* (Vol. 1).
Meyer, G. H. (1998). The numerical valuation of options with underlying jumps. *Acta Math. Univ. Comenianae, 67*(1), 69–82.
Meyer, G. H., & Van der Hoek, J. (1997). The valuation of American options with the method of lines. *Advances in Futures and Options Research, 9,* 265–286.
Mori, M., & Sugihara, M. (2001). The double-exponential transformation in numerical analysis. *Journal of Computational and Applied Mathematics, 127*(1–2), 287–296.
Schiesser, W. E., & Griffiths, G. W. (2009). *A compendium of partial differential equation models: Method of lines analysis with Matlab*.
Strang, G. (2006). *Linear algebra and its applications*.
Volders, K., et al. (2014). Stability and convergence analysis of discretizations of the Black–Scholes PDE with the linear boundary condition. *IMA Journal of Numerical Analysis, 34*(1), 296–325.
Yamamoto, Y. (2005). Double-exponential fast Gauss transform algorithms for pricing discrete lookback options. *Publications of the Research Institute for Mathematical Sciences, 41*(4), 989–1006.