A Sixth-Order CEV Option Valuation Algorithm on Non-uniform Spatial Grids

Nawdha Thakoor(✉)

Department of Mathematics, University of Mauritius, Moka, Mauritius
n.thakoor@uom.ac.mu

Abstract. Due to its ability to fit skew implied volatility profiles of market option prices, the constant elasticity of variance (CEV) model overcomes a shortcoming of the constant volatility assumption used by the Black-Scholes lognormal model. Although the CEV model has an analytical formula for European options, the presence of the non-central Chi-square distribution in the formula brings instability in the computation of European option prices for certain parameter ranges. For efficient valuation of the option prices for all realistic option parameters, a sixth-order finite difference method is proposed in this work. The computational procedure employs a seven-point discretisation of the space derivatives on non-uniform grids together with a mesh-refinement technique around the option's strike price. This new technique is very fast and is shown to achieve sixth-order numerical convergence ratios. The stability of the finite difference method is demonstrated through its capability to generate accurate prices for strongly negative elasticity factors. The numerical method is thus a superior technique to existing methods for numerical computations of CEV option prices.

Keywords: Option valuation · Constant elasticity of variance model · Finite difference algorithm · Non-uniform grids

1 Introduction

Implied volatilities of market option prices across strike prices are not consistent with the assumption of constant volatility in the lognormal dynamics of stock prices. In the CEV model, the constant volatility in the geometric Brownian motion is replaced by a deterministic price-dependent volatility function and the discounted expected payoff of a European style option under the risk-neutral CEV density results in a pricing formula which is in terms of complementary non-central shi-square distribution. When the elasticity parameter in the CEV model is close to a value of one, the computation of option prices becomes very slow [16]. Many authors have considered the efficient computation of the CEV option pricing formula and a comparison of different methods is given in [7].

An alternative to the use of analytical techniques is the computation of option prices by finite difference methods. A fourth-order compact scheme for European options is described in [12] and the methodology of high-order compact discretisation has been

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13377, pp. 435–449, 2022.
https://doi.org/10.1007/978-3-031-10536-4_29

extended to the pricing of different types of barrier options under the CEV dynamics in [13]. An accurate and fast algorithm for CEV American options is described in [14] and localised radial basis functions finite difference (RBF-FD) methods are considered for the pricing of both European and American options in [15]. All these numerical schemes are implicit finite difference methods on uniform grids.

This work proposes a different approach based on explicit finite difference discretisations on non-uniform grids. The method is designed to give significant increases in accuracy over an existing fourth-order accurate method [12] since it achieves sixth-order accuracy, which is two-orders of accuracy higher. The significant gain in accuracy is obtained by employing a non-uniform mesh with grid stretching at the strike price of the European option. The finite difference approximations of the space derivatives in the pricing equations are obtained by interpolation on the non-uniform grid. Analytical formulas for determinants of Vandermonde matrices lead to analytical expressions for the coefficients for approximating the first and second space derivatives. An exponential time stepping scheme [13] leads to fast convergence in time.

Numerical results confirm the sixth-order convergence for options under the Constant Elasticity of Variance (CEV) model. An advantage is that the numerical method remains stable for highly negative values for the parameter of elasticity. This is an important property of the new technique since it is known that, in such cases, low-order computational methods yield inaccurate prices for negative values of the elasticity parameter.

An outline of this work is as follows. The CEV dynamics and the option pricing equation is described in Sect. 2 and explicit finite differences for approximations of the space derivatives in the equation are derived in Sect. 3.

2 The CEV Option Pricing Model

Under the CEV model [2], the stock price dynamics S_t at time $t \geq 0$ follows the stochastic differential equation

$$dS_t = rS_t\,dt + \delta S_t^{\beta}\,dW_t^{\mathbb{Q}}, \tag{1}$$

where β represents the elasticity parameter for the instantaneous local volatility function $\tilde{\sigma}(S) = \delta S^{\beta-1}$ with δ being a positive scale parameter, r is the expected instantaneous return of the stock and W_t, $t \geq 0$ is the standard Brownian motion under $\mathbb{Q}$, the risk-neutral probability measure.

When β is negative, the model (1) produces implied volatility surfaces which are similar to those observed in index options and empirical observations [6] showed that β can become as low as -3 in markets for index options.

The CEV model nests several well-known models as special cases. When $\beta = 1$, the problem reduces to the Black-Scholes model while for $\beta = 1/2$, we obtain the square root diffusion model. Closed form expression for $\beta < 1$ for a European put option with stock price S having maturity T years and strike E are given by

$$V^{\text{exact}} = Ee^{-rT}[1 - \chi^2(d,\, c,\, b)] - S\chi^2(b,\, c+2,\, d), \tag{2}$$

where

$$b = \frac{[Ee^{-rT}]^{2(1-\beta)}}{(1-\beta)^2 f}, \quad c = \frac{1}{1-\beta}, \quad d = \frac{S^{2(1-\beta)}}{(1-\beta^2)f}, \quad f = \frac{\delta^2[e^{2r(\beta-1)T}-1]}{2r(\beta-1)},$$

and $\chi^2(z,\, k,\, l)$ represents the cumulative distribution function (cdf) of a non-central chi-squared random distribution with $k \in \mathbb{N}$ degrees of freedom and $l > 0$ is the non-centrality parameter . It is well know that the evaluation of the non-central χ^2 distribution can become slow as β approaches one, for low volatility problems or small time to maturity and this motivates the use of numerical methods for such options under the CEV model. Fourth-order numerical methods on uniform grids with local mesh refinement have been proposed in [12,13]. In this work, we propose higher-order numerical methods up to order six.

2.1 The CEV PDE

Using no-arbitrage principle, it can be shown that the European option price solves the partial differential equation

$$\frac{\partial V}{\partial t} + \frac{1}{2}\delta^2 S^{2\beta}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \quad t \in [0,\, T], \quad S \geq 0, \tag{3}$$

Pricing a European put option requires (3) using the terminal payoff function $V(S,\, T) = \max(E - S,\, 0)$ with left boundary condition as $V(0,\, t) = Ee^{-rt}$ and right boundary condition given by $V(S,\, t) = 0$ as $S \to +\infty$.

Let the time to maturity be denoted by $\tau = T - t$, then the European put price can be obtained by solving the problem

$$\begin{aligned}
\frac{\partial V}{\partial \tau} &= \frac{1}{2}\delta^2 S^{2\beta}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV, \quad S \geq 0, \quad \tau \in [0,\, T], \\
V(S,\, 0) &= \max(E - S,\, 0), \quad S \geq 0, \\
V(0,\, \tau) &= Ee^{-r\tau}, \quad \tau \in [0,\, T], \\
V(S,\, \tau) &= 0 \text{ as } S \to +\infty, \quad \tau \in [0,\, T].
\end{aligned} \tag{4}$$

3 High-Order Non-uniform Scheme

High-order approximations based on a non-uniform spatial grid is derived using the technique described in [1]

3.1 N^{th}-Order Scheme

Let M be the number of spatial grid nodes and $S_1,\, S_2,\, \ldots,\, S_M$ be the set of distinct non-uniformly spaced grid points. Let $D_{n_p}^{(d)}V(S_m)$ denote the n_p-point approximation for the dth derivative $V^{(d)}(S_m)$ of V at the reference node x_m for $m = 0,\, 1,\, \ldots,\, M$.

Suppose that we require finite difference approximations, $D_{n_p}^{(d)}[V]$, the derivatives of order $(n_p - 1)$ at a given reference node S_m in the form [1]

$$D_{n_p}^{(d)}[V] = \sum_{i=1}^{n_p} w_i V(S_m), \quad 0 \leq d \leq n_p - 1.$$

Using a Taylor series expansion about S_m, we have

$$V(S_m) = \sum_{j=0}^{\infty} \frac{\alpha_{m+k}^j}{j!} \frac{\partial^j V}{\partial S^j}(S_m),$$

where $\alpha_{m+k} = (S_{m+k} - S_m)$ is the displacement of any node S_{m+k} to the reference node S_m where

$$-q \leq k \leq q, \quad \text{with} \quad q = \left\lfloor \frac{n_p}{2} \right\rfloor.$$

Truncating the Taylor series expansion up to order $(n_p - 1)$ gives

$$D_{n_p}^{(d)}[V] = \sum_{k=-q}^{q} w_{m+k}^{(n_p,\,d)} \left(\sum_{j=0}^{n_p-1} \frac{\alpha_{m+k}^j}{j!} \frac{\partial^j f}{\partial S^j}(S_m) \right).$$

Since the above is a finite sum, interchanging the summations we obtain,

$$D_{n_p}^{(d)}[V] = \sum_{j=0}^{n_p-1} \left(\sum_{k=-q}^{q} w_{m+k}^{(n_p,\,d)} \frac{\alpha_{m+k}^j}{j!} \right) \frac{\partial^j V}{\partial S^j}(S_m).$$

For every d, $0 \leq d \leq n_p - 1$, we obtain n_p equations

$$\sum_{k=-q}^{q} \alpha_{m+q}^j w_{m+k}^{(n_p,\,d)} = j! \delta_{d,j}, \quad 0 \leq j \leq n_p - 1.$$

Let $\mathbf{A}$ be the Vandermonde matrix given by

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_{m-q} & \alpha_{m-q+1} & \alpha_{m-q+2} & \cdots & \alpha_{m+q} \\ \alpha_{m-q}^2 & \alpha_{m-q+1}^2 & \alpha_{m-q+2}^2 & \cdots & \alpha_{m+q}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{m-q}^{n_p-1} & \alpha_{m-q+1}^{n_p-1} & \alpha_{m-q+2}^{n_p-1} & \cdots & \alpha_{m+q}^{n_p-1} \end{pmatrix}.$$

Thus for every d, $0 \leq d \leq n_p - 1$, we have to solve the matrix equation

$$\mathbf{A}\mathbf{w}^{(n_p,\,d)} = \mathbf{b}_d,$$

where

$$\mathbf{w}^{(n_p,\,d)} = \left[w_{m-q}^{(n_p,\,d)}, \, w_{m-q+1}^{(n_p,\,d)}, \, \ldots, \, w_{m+q}^{(n_p,\,d)} \right]^T,$$

$$\mathbf{b}_d = \left[\delta_{d,0},\, \delta_{d,1},\, 2!\delta_{d,2},\, 3!\delta_{d,3},\, \ldots,\, (n_p-1)!\delta_{d,n_p-1}\right]^T,$$

and the Kronecker delta is defined as

$$\delta_{i,j} = \begin{cases} 1, & i = j; \\ 0, & i \neq j. \end{cases}$$

Since

$$\det \mathbf{A} = \prod_{1 \le i < j \le n_p} (\alpha_i - \alpha_j)\,,$$

using Cramer's rule, we find that

$$w_{m+k}^{(n_p,\,d)} = \frac{-\det \begin{pmatrix} \mathbf{A} & \mathbf{b}_d \\ \mathbf{c}(k) & 0 \end{pmatrix}}{\det \mathbf{A}}, \quad -q \le k \le q,$$

where

$$\mathbf{c}(k) = \left[\delta_{m-q,\,m+k},\, \delta_{m-q+1,\,m+k},\, \ldots,\, \delta_{m,\,m+k},\, \delta_{m+1,\,m+k},\, \ldots,\, \delta_{m+q,\,m+k}\right].$$

Second-Order Approximations. For an $n_p = 3$-point stencil approximation centered at the reference node S_m, we can obtain second-order approximate results and we require the solution of

$$w_{m+k}^{(3,d)} = \frac{-\det \begin{pmatrix} 1 & 1 & 1 & \delta_{d,0} \\ \alpha_{m-1} & \alpha_m & \alpha_{m+1} & \delta_{d,1} \\ \alpha_{m-1}^2 & \alpha_m^2 & \alpha_{m+1}^2 & 2!\delta_{d,2} \\ \delta_{m-1,\,m+k} & \delta_{m,\,m+k} & \delta_{m+1,\,m+k} & 0 \end{pmatrix}}{\det \begin{pmatrix} 1 & 1 & 1 \\ \alpha_{m-1} & \alpha_m & \alpha_{m+1} \\ \alpha_{m-1}^2 & \alpha_m^2 & \alpha_{m+1}^2 \end{pmatrix}}, \quad k = -1,\, 0,\, 1.$$

The weights for the first derivative $(d = 1)$ and the second derivative $(d = 2)$ are given in the Table 1.

Table 1. Weights for $n_p = 3$ points

weights for first derivative $(d = 1)$	weights for second Derivative $(d = 2)$
$w_{m-1}^{(3,\,1)} = -\frac{\alpha_m + \alpha_{m+1}}{(\alpha_{m-1}-\alpha_m)(\alpha_{m-1}-\alpha_{m+1})}$	$w_{m-1}^{(3,\,2)} = \frac{2}{(\alpha_{m-1}-\alpha_{m+1})(\alpha_{m-1}-\alpha_m)}$
$w_m^{(3,\,1)} = \frac{\alpha_{m-1}+\alpha_{m+1}}{(\alpha_{m-1}-\alpha_m)(\alpha_m-\alpha_{m+1})}$	$w_m^{(3,\,2)} = \frac{2}{(\alpha_m-\alpha_{m+1})(\alpha_m-\alpha_{m-1})}$
$w_{m+1}^{(3,\,1)} = \frac{\alpha_{m-1}+\alpha_m}{(\alpha_{m-1}-\alpha_{m+1})(\alpha_{m+1}-\alpha_m)}$	$w_{m+1}^{(3,\,2)} = \frac{2}{(\alpha_{m+1}-\alpha_m)(\alpha_{m+1}-\alpha_{m-1})}$

The matrices for the derivatives of order one and two for $D_{n_p}^{(d)}[V]$ are then constructed as

$$D_3^{(1)} = \begin{pmatrix} \tilde{\omega}_{1+}^{(3,1)} & \tilde{\omega}_{2+}^{(3,1)} & \tilde{\omega}_{3+}^{(3,1)} & & \\ w_0^{(3,1)} & w_1^{(3,1)} & w_2^{(3,1)} & & \\ \ddots & \ddots & \ddots & & \\ & w_{M-2}^{(3,1)} & w_{M-1}^{(3,1)} & w_M^{(3,1)} & \\ & \tilde{\omega}_{-3}^{(3,1)} & \tilde{\omega}_{-2}^{(3,1)} & \tilde{\omega}_{-1}^{(3,1)} & \end{pmatrix}, \quad D_3^{(2)} = \begin{pmatrix} \tilde{\omega}_{1+}^{(3,2)} & \tilde{\omega}_{2+}^{(3,2)} & \tilde{\omega}_{3+}^{(3,2)} & \tilde{\omega}_{4+}^{(3,2)} & \\ w_0^{(3,2)} & w_1^{(3,2)} & w_2^{(3,2)} & & \\ \ddots & \ddots & \ddots & & \\ & w_{M-2}^{(3,2)} & w_{M-1}^{(3,2)} & w_M^{(3,2)} & \\ \tilde{\omega}_{-4}^{(3,2)} & \tilde{\omega}_{-3}^{(3,2)} & \tilde{\omega}_{-2}^{(3,2)} & \tilde{\omega}_{-1}^{(3,2)} & \end{pmatrix}. \quad (5)$$

Note that at the left boundary S_0, instead of using central approximations, we use one-sided approximations and we expand at the reference node S_0 where $\alpha_i = S_i - S_0$, $i = 0, 1, 2$, which gives

$$\left[\tilde{\omega}_{1+}^{(3,1)}, \tilde{\omega}_{2+}^{(3,1)}, \tilde{\omega}_{3+}^{(3,1)}\right] = \left[\frac{\alpha_2 - \alpha_1}{\alpha_1\alpha_2}, \frac{\alpha_2}{\alpha_1(\alpha_2 - \alpha_1)}, \frac{\alpha_1}{\alpha_2(\alpha_1 - \alpha_2)}\right].$$

Similarly at the right boundary, we approximate at the reference node S_M where $\alpha_i = S_i - S_M$, $i = M-2, M-1, M$, with $\alpha_M = 0$, which gives

$$\left[\tilde{\omega}_{-3}^{(3,1)}, \tilde{\omega}_{-2}^{(3,1)}, \tilde{\omega}_{-1}^{(3,1)}\right] = \left[-\frac{\alpha_{M-1}}{\alpha_{M-2}(\alpha_{M-2}\alpha_{M-1})}, \frac{\alpha_{M-2} - \alpha_{M-1}}{\alpha_{M-1}(\alpha_{M-1}\alpha_{M-2})}, -\frac{\alpha_{M-2} + \alpha_{M-1}}{\alpha_{M-2}\alpha_{M-1}}\right].$$

The same procedure with 4 points instead of 3, is adopted for weights corresponding to the second derivatives and the one-sided coefficients at the nodes S_0 and S_M are given by

$$\left[\tilde{\omega}_{1+}^{(3,2)}, \tilde{\omega}_{2+}^{(3,2)}, \tilde{\omega}_{3+}^{(3,2)}, \tilde{\omega}_{4+}^{(3,2)}\right] =$$
$$\left[\frac{2(\alpha_1 + \alpha_2 + \alpha_3)}{\alpha_1\alpha_2\alpha_3}, \frac{-2(\alpha_2 + \alpha_3)}{\alpha_1(\alpha_1 - \alpha_3)(\alpha_1 - \alpha_2)}, \frac{-2(\alpha_1 + \alpha_3)}{\alpha_2(\alpha_2 - \alpha_3)(\alpha_2 - \alpha_1)}, \frac{-2(\alpha_1 + \alpha_2)}{\alpha_3(\alpha_3 - \alpha_1)(\alpha_3 - \alpha_2)},\right]$$

and

$$\tilde{\omega}_{-4}^{(3,2)} = \frac{-2(\alpha_{M-2} + \alpha_{M-1})}{\alpha_{M-3}(\alpha_{M-3} - \alpha_{M-2})(\alpha_{M-3} - \alpha_{M-1})}, \quad \tilde{\omega}_{-3}^{(3,2)} = \frac{-2(\alpha_{M-3} + \alpha_{M-1})}{\alpha_{M-2}(\alpha_{M-2} - \alpha_{M-3})(\alpha_{M-2} - \alpha_{M-1})}$$

$$\tilde{\omega}_{-2}^{(3,2)} = \frac{-2(\alpha_{M-3} + \alpha_{M-2})}{\alpha_{M-1}(\alpha_{M-1} - \alpha_{M-3})(\alpha_{M-1} - \alpha_{M-2})}, \quad \tilde{\omega}_{-1}^{(3,2)} = \frac{2(\alpha_{M-3} + \alpha_{M-2} + \alpha_{M-1})}{\alpha_{M-3}\alpha_{M-2}\alpha_{M-1}}.$$

Fourth-Order Approximations. With an $n_p = 5$-point stencil approximation centered at the reference node S_m and taking a fourth-order Taylor series expansion of V gives the approximation for the dth derivative as

$$D_5^{(d)}[V] = \sum_{k=-2}^{2} \left(w_{m+k}^{(5,d)} \left(\sum_{j=0}^{\infty} \frac{\alpha_{m+k}^j}{j!} \frac{\partial^j V}{\partial S^j}(S_m) \right) \right).$$

The weights are then obtained by solving

$$w_{m+k}^{(5,d)} = \frac{-\det\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \delta_{d,0} \\ \alpha_{m-2} & \alpha_{m-1} & \alpha_m & \alpha_{m+1} & \alpha_{m+2} & \delta_{d,1} \\ \alpha_{m-2}^2 & \alpha_{m-1}^2 & \alpha_m^2 & \alpha_{m+1}^2 & \alpha_{m+2}^2 & 2!\delta_{d,2} \\ \alpha_{m-2}^3 & \alpha_{m-1}^3 & \alpha_m^3 & \alpha_{m+1}^3 & \alpha_{m+2}^3 & 3!\delta_{d,3} \\ \alpha_{m-2}^4 & \alpha_{m-1}^4 & \alpha_m^4 & \alpha_{m+1}^4 & \alpha_{m+2}^4 & 4!\delta_{d,4} \\ \delta_{m-2,\,m+k} & \delta_{m-1,\,m+k} & \delta_{m,\,m+k} & \delta_{m+1,\,m+k} & \delta_{m+2,\,m+k} & 0 \end{pmatrix}}{|\mathbf{A}_5|},$$

for $k = -2, -1, 0, 1, 2$, where $|\mathbf{A}_5|$ is the determinant of the toeplitz matrix corresponding to $n_p = 5$ points. The weights for the first derivative are then given by

$$w_{m-2}^{(5,\,1)} = -\frac{\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}\left(\alpha_{m+1}\alpha_{m+2} + \alpha_m\left(\alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m-1}\right)\left(\alpha_{m-2} - \alpha_m\right)\left(\alpha_{m-2} - \alpha_{m+1}\right)\left(\alpha_{m-2} - \alpha_{m+2}\right)},$$

$$w_{m-1}^{(5,\,1)} = \frac{\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-2}\left(\alpha_{m+1}\alpha_{m+2} + \alpha_m\left(\alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m-1}\right)\left(\alpha_{m-1} - \alpha_m\right)\left(\alpha_{m-1} - \alpha_{m+1}\right)\left(\alpha_{m-1} - \alpha_{m+2}\right)},$$

$$w_{m}^{(5,\,1)} = \frac{\alpha_{m-1}\alpha_{m+1}\alpha_{m+2} + \alpha_{m-2}\left(\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}\left(\alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_m\right)\left(\alpha_m - \alpha_{m-1}\right)\left(\alpha_m - \alpha_{m+1}\right)\left(\alpha_m - \alpha_{m+2}\right)}$$

$$w_{m+1}^{(5,\,1)} = \frac{\alpha_{m-1}\alpha_m\alpha_{m+2} + \alpha_{m-2}\left(\alpha_m\alpha_{m+2} + \alpha_{m-1}\left(\alpha_m + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m+1}\right)\left(\alpha_{m+1} - \alpha_{m-1}\right)\left(\alpha_{m+1} - \alpha_m\right)\left(\alpha_{m+1} - \alpha_{m+2}\right)},$$

$$w_{m+2}^{(5,\,1)} = \frac{\alpha_{m-1}\alpha_m\alpha_{m+1} + \alpha_{m-2}\left(\alpha_m\alpha_{m+1} + \alpha_{m-1}\left(\alpha_m + \alpha_{m+1}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m+2}\right)\left(\alpha_{m+2} - \alpha_{m-1}\right)\left(\alpha_{m+2} - \alpha_m\right)\left(\alpha_{m+2} - \alpha_{m+1}\right)},$$

and those for the second derivatives are given by

$$w_{m-2}^{(5,\,2)} = \frac{2\left(\alpha_{m+1}\alpha_{m+2} + \alpha_m\left(\alpha_{m+1} + \alpha_{m+2}\right) + \alpha_{m-1}\left(\alpha_m + \alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m-1}\right)\left(\alpha_{m-2} - \alpha_m\right)\left(\alpha_{m-2} - \alpha_{m+1}\right)\left(\alpha_{m-2} - \alpha_{m+2}\right)},$$

$$w_{m-1}^{(5,\,2)} = \frac{2\left(\alpha_{m+1}\alpha_{m+2} + \alpha_m\left(\alpha_{m+1} + \alpha_{m+2}\right) + \alpha_{m-2}\left(\alpha_m + \alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-1} - \alpha_{m-2}\right)\left(\alpha_{m-1} - \alpha_m\right)\left(\alpha_{m-1} - \alpha_{m+2}\right)\left(\alpha_{m-1} - \alpha_{m+1}\right)},$$

$$w_{m}^{(5,\,2)} = -\frac{2\left(\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}\left(\alpha_{m+1} + \alpha_{m+2}\right) + \alpha_{m-2}\left(\alpha_{m-1} + \alpha_{m+1} + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_m\right)\left(\alpha_m - \alpha_{m-1}\right)\left(\alpha_m - \alpha_{m+1}\right)\left(\alpha_m - \alpha_{m+2}\right)},$$

$$w_{m+1}^{(5,\,2)} = -\frac{2\left(\alpha_m\alpha_{m+2} + \alpha_{m-1}\left(\alpha_m + \alpha_{m+2}\right) + \alpha_{m-2}\left(\alpha_{m-1} + \alpha_m + \alpha_{m+2}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m+1}\right)\left(\alpha_{m+1} - \alpha_{m-1}\right)\left(\alpha_{m+1} - \alpha_m\right)\left(\alpha_{m+1} - \alpha_{m+2}\right)},$$

$$w_{m+2}^{(5,\,2)} = -\frac{2\left(\alpha_m\alpha_{m+1} + \alpha_{m-1}\left(\alpha_m + \alpha_{m+1}\right) + \alpha_{m-2}\left(\alpha_{m-1} + \alpha_m + \alpha_{m+1}\right)\right)}{\left(\alpha_{m-2} - \alpha_{m+2}\right)\left(\alpha_{m+2} - \alpha_{m-1}\right)\left(\alpha_{m+2} - \alpha_m\right)\left(\alpha_{m+2} - \alpha_{m+1}\right)}.$$

The derivatives matrices for $D_{n_p}^{(d)}[V]$ using the 5-point stencil are then constructed as in (5) in the form

$$D_5^{(1)} = \text{diag}\left[w_{m-2}^{(3,\,1)},\, w_{m-1}^{(3,\,1)},\, w_m^{(3,\,1)},\, w_{m+1}^{(3,\,1)},\, w_{m+2}^{(3,\,1)}\right],$$

$$D_5^{(2)} = \text{diag}\left[w_{m-2}^{(3,\,1)},\, w_{m-1}^{(3,\,2)},\, w_m^{(3,\,2)},\, w_{m+1}^{(3,\,2)},\, w_{m+2}^{(3,\,2)}\right],$$

for $2 \le m \le M-2$ and $D_5^{(d)} \in \mathbb{R}^{(M+1)\times(M+1)}$.

For the 5-point scheme, one sided approximations are required at left ends S_0, S_1 and the right ends S_{M-1}, S_M of the computational domain. The coefficients can be determined using a similar way as above using the displacements $\alpha_i = S_i - S_j$ for $j = 0,\ 1,\ M-2,\ M-1$. For instance, for the first row of $D_5^{(1)}$ which corresponds to grid node S_0, the displacements is given by

$$\{\alpha_0,\ \alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4\} = \{0,\ S_1 - S_0,\ S_2 - S_0,\ S_3 - S_0,\ S_4 - S_0\},$$

and the weights are calculated accordingly. Similarly the entries of the second row of $D_5^{(1)}$ evaluated at S_1 are obtained by calculating the weights corresponding to the displacements

$$\{\alpha_0,\ \alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4\} = \{S_0 - S_1,\ 0,\ S_2 - S_1,\ S_3 - S_1,\ S_4 - S_1\}.$$

The one-sided approximations for the 5-point second derivative matrix is calculated in a similar way.

Sixth-Order Approximations. To obtain sixth order approximations, we use an $n_p = 7$-point stencil approximation centered at the reference node S_m and proceeding in the same way we obtain the weights for the first derivative as

$$w_{m-3}^{(7,1)} = -\frac{\begin{array}{c}\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-2}(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+3} + \alpha_{m+2}))))\end{array}}{\begin{array}{c}(\alpha_{m-3} - \alpha_{m-2})\,(\alpha_{m-3} - \alpha_{m-1})\,(\alpha_{m-3} - \alpha_m) \\ (\alpha_{m-3} - \alpha_{m+1})\,(\alpha_{m-3} - \alpha_{m+3})\,(\alpha_{m-3} - \alpha_{m+2})\end{array}},$$

$$w_{m-2}^{(7,1)} = \frac{\begin{array}{c}\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-3}(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+3} + \alpha_{m+2}))))\end{array}}{\begin{array}{c}(\alpha_{m-3} - \alpha_{m-2})\,(\alpha_{m-2} - \alpha_{m-1})\,(\alpha_{m-2} - \alpha_m) \\ (\alpha_{m-2} - \alpha_{m+2})\,(\alpha_{m-2} - \alpha_{m+1})\,(\alpha_{m-2} - \alpha_{m+3})\end{array}},$$

$$w_{m-1}^{(7,1)} = -\frac{\begin{array}{c}\alpha_{m-2}\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-3}(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{c}(\alpha_{m-1} - \alpha_{m-2})\,(\alpha_{m-1} - \alpha_{m-3})\,(\alpha_{m-1} - \alpha_m) \\ (\alpha_{m-1} - \alpha_{m+1})\,(\alpha_{m-1} - \alpha_{m+3})\,(\alpha_{m-1} - \alpha_{m+2})\end{array}},$$

$$w_m^{(7,1)} = \frac{\begin{array}{c}\alpha_{m-2}\alpha_{m-1}\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-3}(\alpha_{m-1}\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+3} + \alpha_{m+2}))))\end{array}}{\begin{array}{c}(\alpha_m - \alpha_{m+1})\,(\alpha_{m-3} - \alpha_m)\,(\alpha_m - \alpha_{m-2})\,(\alpha_m - \alpha_{m+3}) \\ (\alpha_m - \alpha_{m+2})\,(\alpha_m - \alpha_{m-1})\end{array}},$$

$$w_{m+1}^{(7,1)} = \frac{\begin{array}{c}\alpha_{m-2}\alpha_{m-1}\alpha_m\alpha_{m+2}\alpha_{m+3} + \alpha_{m-3}(\alpha_{m-1}\alpha_m\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m-2}(\alpha_m\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{c}(\alpha_{m-3} - \alpha_{m+1})\,(\alpha_{m+1} - \alpha_{m-2})\,(\alpha_{m+1} - \alpha_{m-1}) \\ (\alpha_{m+1} - \alpha_m)\,(\alpha_{m+1} - \alpha_{m+3})\,(\alpha_{m+1} - \alpha_{m+2})\end{array}},$$

$$w_{m+2}^{(7,1)} = \frac{\begin{array}{c}\alpha_{m-2}\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+3} + \alpha_{m-3}(\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+3} \\ + \alpha_{m-2}(\alpha_m\alpha_{m+1}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+3} + \alpha_m(\alpha_{m+1} + \alpha_{m+3}))))\end{array}}{\begin{array}{c}(\alpha_{m+2} - \alpha_m)\,(\alpha_{m+2} - \alpha_{m-2})\,(\alpha_{m+2} - \alpha_{m-1}) \\ (\alpha_{m-3} - \alpha_{m+2})\,(\alpha_{m+2} - \alpha_{m+1})\,(\alpha_{m+2} - \alpha_{m+3})\end{array}},$$

$$w_{m+3}^{(7,1)} = \frac{\begin{array}{r}\alpha_{m-2}\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-3}(\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+2} \\ + \alpha_{m-2}(\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2} + \alpha_m(\alpha_{m+2} + \alpha_{m+1}))))\end{array}}{\begin{array}{r}(\alpha_{m+3} - \alpha_{m+2})\,(\alpha_{m-3} - \alpha_{m+3})\,(\alpha_{m+3} - \alpha_{m-2})\,(\alpha_{m+3} - \alpha_{m-1}) \\ (\alpha_{m+3} - \alpha_{m+1})\,(\alpha_{m+3} - \alpha_m)\end{array}}.$$

The weights for the second derivative are given by

$$w_{m-3}^{(7,2)} = \frac{\begin{array}{l}2(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}))) + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3})) + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}) \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_{m-3} - \alpha_{m-2})(\alpha_{m-3} - \alpha_{m-1})(\alpha_{m-3} - \alpha_m) \\ (\alpha_{m-3} - \alpha_{m+1})(\alpha_{m-3} - \alpha_{m+2})(\alpha_{m-3} - \alpha_{m+3})\end{array}}$$

$$w_{m-2}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+1}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}))) + \alpha_{m-3}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3})) + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}) \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_{m-2} - \alpha_{m-1})(\alpha_{m-2} - \alpha_m)(\alpha_{m-3} - \alpha_{m-2})(\alpha_{m-2} - \alpha_{m+1}) \\ (\alpha_{m-2} - \alpha_{m+3})(\alpha_{m-2} - \alpha_{m+2})\end{array}}$$

$$w_{m-1}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_m\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-2}(\alpha_{m+3}\alpha_{m+1}\alpha_{m+2} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}))) + \alpha_{m-3}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3})) + \alpha_{m-2}(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}) \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_{m-3} - \alpha_{m-1})(\alpha_{m-1} - \alpha_{m-2})(\alpha_{m-1} - \alpha_m) \\ (\alpha_{m-1} - \alpha_{m+1})(\alpha_{m-1} - \alpha_{m+2})(\alpha_{m-1} - \alpha_{m+3})\end{array}}$$

$$w_{m}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_{m-1}\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}))) + \alpha_{m-3}(\alpha_{m+1}\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3})) + \alpha_{m-2}(\alpha_{m+2}\alpha_{m+3} + \alpha_{m+1}(\alpha_{m+2} + \alpha_{m+3}) \\ + \alpha_{m-1}(\alpha_{m+1} + \alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_m - \alpha_{m-2})(\alpha_m - \alpha_{m-1})(\alpha_m - \alpha_{m+1}) \\ (\alpha_m - \alpha_{m+2})(\alpha_{m-3} - \alpha_m)(\alpha_m - \alpha_{m+3})\end{array}}$$

$$w_{m+1}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_{m-1}\alpha_m\alpha_{m+2}\alpha_{m+3} + \alpha_{m-2}(\alpha_m\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_m(\alpha_{m+2} + \alpha_{m+3}))) + \alpha_{m-3}(\alpha_m\alpha_{m+2}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+2}\alpha_{m+3} \\ + \alpha_m(\alpha_{m+2} + \alpha_{m+3})) + \alpha_{m-2}(\alpha_{m+2}\alpha_{m+3} + \alpha_m(\alpha_{m+2} + \alpha_{m+3}) \\ + \alpha_{m-1}(\alpha_m + \alpha_{m+2} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_{m-3} - \alpha_{m+1})(\alpha_{m+1} - \alpha_{m-2})(\alpha_{m+1} - \alpha_{m-1}) \\ (\alpha_{m+1} - \alpha_m)(\alpha_{m+1} - \alpha_{m+2})(\alpha_{m+1} - \alpha_{m+3})\end{array}}$$

$$w_{m+2}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+3} + \alpha_{m-2}(\alpha_m\alpha_{m+1}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+3} \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+3}))) + \alpha_{m-3}(\alpha_m\alpha_{m+1}\alpha_{m+3} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+3} \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+3})) + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+3} + \alpha_m(\alpha_{m+1} + \alpha_{m+3}) \\ + \alpha_{m-1}(\alpha_m + \alpha_{m+1} + \alpha_{m+3}))))\end{array}}{\begin{array}{r}(\alpha_{m+2} - \alpha_{m-1})(\alpha_{m-3} - \alpha_{m+2})(\alpha_{m+2} - \alpha_{m-2}) \\ (\alpha_{m+2} - \alpha_m)(\alpha_{m+2} - \alpha_{m+1})(\alpha_{m+2} - \alpha_{m+3})\end{array}}$$

$$w_{m+3}^{(7,2)} = -\frac{\begin{array}{l}2(\alpha_{m-1}\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-2}(\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2} \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+2}))) + \alpha_{m-3}(\alpha_m\alpha_{m+1}\alpha_{m+2} + \alpha_{m-1}(\alpha_{m+1}\alpha_{m+2} \\ + \alpha_m(\alpha_{m+1} + \alpha_{m+2})) + \alpha_{m-2}(\alpha_{m+1}\alpha_{m+2} + \alpha_m(\alpha_{m+1} + \alpha_{m+2}) \\ + \alpha_{m-1}(\alpha_m + \alpha_{m+1} + \alpha_{m+2}))))\end{array}}{\begin{array}{r}(\alpha_{m-3} - \alpha_{m+3})(\alpha_{m+3} - \alpha_{m-2})(\alpha_{m+3} - \alpha_{m-1}) \\ (\alpha_{m+3} - \alpha_m)(\alpha_{m+3} - \alpha_{m+1})(\alpha_{m+3} - \alpha_{m+2})\end{array}}$$

At the left and right boundaries, the weights are calculated as described earlier. The Mathematica code used to obtain the weights is given in Fig. 1.

```
(* Input Parameters *)
np = 7; d = 1;
(* Construction of the VanderMonde Matrix *)
q = Floor[np / 2];
list = Table[α_k, {k, m - q, m + q}];
vandermonde[a_] := Transpose[Outer[Power, a, Range[0, Length[a] - 1]]]
A = vandermonde[list];
(* Obtaining the weights for the derivative matrix of order d *)
DetA = Simplify[Det[A]];
b = ConstantArray[0, np]; b[[d + 1]] = d!;
row[v_] := {v}; col[v_] := {#} & /@ v; sca[x_] := {{x}}
i = 0;
For[k = -q, k <= q, k++,
 i++;
 c = ConstantArray[0, np]; c[[i]] = 1;
 AugA_i = ArrayFlatten[{{A, col[b]}, {row[c], sca[0]}}];
 w_i = Simplify[1/DetA * -Simplify[Det[AugA_i]]]; Print[w_{m+k}, "=", w_i]]
```

Fig. 1. Mathematica code to derive the weights for the derivative matrices using n_p-point stencil.

4 Applications to Option Pricing

We next consider an application of the high-order method in §3 to option pricing when the asset price follows a CEV process.

4.1 Numerical Discretisation of the CEV PDE

We localise the problem (4) to a finite computational domain $\Omega = [S_{\min}, S_{\max}] \times [0, T]$. The non-uniform spatial grid is constructed as in Sect. 4.2. Let $V_m(\tau) = V(S_m, \tau)$ and let $\mathbf{V}(\tau) = [V_0(\tau), V_1(\tau), \ldots, V_M(\tau)]^T$. The pde (4) becomes

$$\mathbf{V}'(\tau) = \mathbf{L}\mathbf{V}(\tau), \tag{6}$$

where $\mathbf{L} \in \mathbb{R}^{(M+1)\times(M+1)}$ is given by

$$\mathbf{L} = \frac{1}{2}\delta^2 \mathbf{S}_I^{2\beta} \mathbf{D}_{n_p}^{(2)} + r\mathbf{S}_I \mathbf{D}_{n_p}^{(1)} - r\mathbf{I},$$

where the first order and second order derivative matrices $\mathbf{D}_{n_p}^{(1)}$ and $\mathbf{D}_{n_p}^{(2)}$ are constructed using the weights derived in Sect. 3.1 for $n_p \in \{3, 5, 7\}$, $\mathbf{I}$ represents the $(M+1) \times (M+1)$ identity matrix and

$$\mathbf{S}_I = \text{diag}[S_0, S_1, \ldots, S_M] \quad \in \mathbb{R}^{(M+1)\times(M+1)}.$$

An exponential time integration [9] of (6) between 0 and T gives

$$\mathbf{V}(T) = e^{\mathbf{L}T}\mathbf{V}(0), \tag{7}$$

where $\mathbf{V}(0) = [(E-S_0)^+, (E-S_1)^+, \ldots, (E-S_M)^+]$ with $(E-S_m)^+ = \max(E-S_m, 0)$. The evaluation of the exponential matrix in (7) can be efficiently performed by the Carathéodory-Fejér approximation and contour integrals [8].

For the pricing of the more complex American options, we employ an operator splitting strategy [5] for solving the linear complementarity problem [11] as employed in the case of a 5-point stencil approximation using a localised radial point interpolation method [10].

4.2 Grid-Spacing

Consider the finite computational domain $[S_{\min}, S_{\max}] \times [0, T]$. Due to the non-smooth payoff function for option pricing problems which hampers the rate of convergence of high-order methods, a grid stretching strategy near the payoff's kink E is required to restore numerical solutions exhibiting high-order convergence rates. We therefore employ a grid stretching strategy to concentrate more grid nodes at E which restores the expected high-order rate of convergence with the effect of decreasing the error at the point of discontinuity. Let $y \in [0, 1]$ be the transformed grid with $y_m = mh$ where $h = 1/M$, then

$$S_m = E + \frac{1}{\xi}\sinh\left(c_1 + (c_2 - c_1)y_m\right),$$

where

$$c_1 = \sinh^{-1}\left(\xi(S_{\min} - E)\right), \quad c_2 = \sinh^{-1}\left(\xi(S_{\max} - E)\right),$$

and ξ is a stretching parameter.

5 Numerical Experiments

We present some numerical examples for pricing options using the high-order non-uniform methods under CEV model which also nests the popular lognormal Black-Scholes model for $\beta = 1$. All computations have been carried out using MATLAB R2020 on a Core i5 laptop with 8 GB RAM and speed 2.60 GHZ.

Our first example prices a $T = 1$ year European put option when the stock has an initial price $S^0 = 100$. The other parameters are $\tilde{\sigma} = 0.1$, strike price $E = 100$ and $r = 0.05$. The corresponding results are shown in Table 2.

The errors over the whole computational domain are computed as

$$||e||_{\ell_2} = \sqrt{\frac{\sum_{j=0}^{M} \left| V_j^{\text{approx}}(T) - V_j^{\text{exact}}(T) \right|^2}{M+1}},$$

and the rate of convergence is obtained from

$$\text{Order} = \log_2 \left(\frac{||e(M)||_{\ell_2}}{||e(2M)||_{\ell_2}} \right).$$

Table 2. Errors and convergence in the ℓ_2-norm under the lognormal Black-Scholes model

$\beta = 1$, $S^0 = 100$, $E = 100$, $\tilde{\sigma} = 0.1$, $r = 0.05$, $T = 1$, $\xi = 80$									
	3-point			5-point			7-point		
M	$\|\|e\|\|_{\ell_2}$	Order	Cpu(s)	$\|\|e\|\|_{\ell_2}$	Order	Cpu(s)	$\|\|e\|\|_{\ell_2}$	Order	Cpu(s)
20	2.1e−1	–	0.009	1.5e−1	–	0.006	8.0e−1	–	0.005
40	5.3e−2	1.998	0.012	1.1e−2	3.790	0.012	8.1e−3	6.626	0.011
80	1.3e−2	1.989	0.014	7.6e−4	3.832	0.016	1.3e−4	5.997	0.013
160	3.3e−3	1.997	0.018	4.9e−5	3.956	0.019	2.2e−6	5.863	0.016
320	8.4e−4	1.999	0.024	3.1e−6	3.989	0.023	3.7e−8	5.883	0.018
640	2.1e−4	2.000	0.031	1.9e−7	3.994	0.035	3.5e−10	6.717	0.022
Exact price = 6.80474876423									

It is observed that the expected convergence rates of order 2, 4 and 6 with the 3, 5 and 7 point stencil schemes are achieved respectively. While the 3-point and 5-point schemes yield a numerical solution having an error of magnitude 10^{-4} and 10^{-6} in around 22 ms, the 7-point scheme leads to an error of magnitude 10^{-10} in the same computational time.

We next consider the pricing of a one-year European put option where the stock price follows the CEV dynamics. The exact price is obtained using (2) and the errors are calculated at the spot price $S^0 = 100$. We choose $\beta = -2$ and other parameters

are listed in Table 3. The option prices for varying M, errors, convergence rates and the computational times are given in Table 3. It can be observed that highly accurate prices are obtained in little computational time.

Table 3. Price, errors and convergence rates under CEV model.

$\beta = -2,\ E = 100,\ S^0 = 100,\ \tilde{\sigma} = 0.1,\ \xi = 70,\ r = 0.05,\ T = 1$												
	3-point				5-point				7-point			
M	Price	Error	Order	Cpu(s)	Price	Error	Order	Cpu(s)	Price	Error	Order	Cpu(s)
20	6.581204	2.2e−1	–	0.006	6.940188	1.3e−1	–	0.005	6.548769	2.6e−1	–	0.006
40	6.754252	5.8e−2	1.995	0.009	6.822525	1.0e−2	3.629	0.009	6.807916	4.3e−3	5.950	0.008
80	6.797551	1.5e−2	1.986	0.013	6.812904	7.3e−4	3.827	0.014	6.812094	8.1e−5	5.711	0.011
160	6.808510	3.7e−3	1.996	0.016	6.812222	4.7e−5	3.955	0.017	6.812174	1.4e−6	5.859	0.013
320	6.811258	9.2e−4	1.999	0.021	6.812178	3.0e−6	3.989	0.021	6.812175	2.4e−8	5.841	0.018
640	6.811946	2.3e−4	2.000	0.029	6.812175	1.9e7	3.994	0.029	6.812175	3.8e−10	5.989	0.020
Exact price = 6.81217523240787												

In Table 4, we show that the proposed technique also yields accurate results for highly negative values of the elasticity parameter. The prices and errors for different values of β with different strike prices are reported in the Table 4.

Table 4. Highly negative values of the elasticity parameter under CEV model.

$S^0 = 100,\ T = 1,\ \xi = 70,\ M = 320,\ \tilde{\sigma} = 0.1,\ r = 0.05$						
	$E = 90$		$E = 100$		$E = 110$	
β	Price	Error	Price	Error	Price	Error
−3	14.851632	2.7e−8	6.821461	2.3e−8	1.703848	4.0e−7
−4	14.952419	2.8e−8	6.835034	2.4e−8	1.568522	3.9e−7
−5	15.073118	3.1e−8	6.853756	3.1e−8	1.442271	3.9e−7
−6	15.218544	3.7e−8	6.879111	4.3e−8	1.324391	3.8e−7
−7	15.383637	4.0e−8	6.912053	5.7e−8	1.214294	4.4e−7

Finally the last numerical example illustrate the effectiveness of the proposed method for pricing American options for which no simple closed-form formula exists. Comparisons is carried out against the Han-Wu algorithm [4] and a RBF-FD method [3]. The results obtained using the 5-point scheme, interpolated at the initial stock price $S^0 = 100$ are shown in Table 5. It is observed that the results obtained using 5-point scheme agree well with the well-known benchmark prices which are available in the literature.

Table 5. American options using the 5-point scheme with $M = 1000$.

$\beta = 1,\ r = 0.05,\ E = 100,\ T = 0.5,\ \tilde{\sigma} = 0.2$							
	Han Wu [4]		RBF-FD [3]		5-point		
S_0	Price	Error	Price	Error	Price	Error	Benchmark
80	20.00000	2.0e−7	20.00000	1.2e−6	20.00000	4.1e−7	20.00000
90	10.66603	3.0e−6	10.66009	5.9e−3	10.66605	4.4e−5	10.66610
100	4.65552	1.0e−4	4.65167	4.0e−3	4.65563	4.2e−5	4.65568
110	1.66797	4.7e−6	1.66622	1.8e−3	1.66798	7.7e−6	1.66799
120	0.49758	2.1e−5	0.49714	4.2e−4	0.49756	1.3e−5	0.49756

6 Conclusion

In this work, we proposed a new sixth-order method for pricing options under the stock price dynamics with heteroscedasticity. The capability to compute accurate prices for strongly negative elasticity parameters shows the merit of the new method.

Acknowledgement. The author gratefully acknowledges the funding support for a high-performance laptop under the award number ID-2019-14 from the Higher Education Commission, Mauritius, which was used for the symbolic derivations in Mathematica and numerical computations in Matlab.

References

1. Bowen, M.K., Smith, R.: Derivative formulae and errors for non-uniformly spaced points. Math. Phys. Eng. Sci. **461**, 1975–1997 (2005)
2. Cox, J.: The constant elasticity of variance option pricing model. J. Portf. Manag. **22**, 16–17 (1996)
3. Golbabai, A., Mohebianfar, E.: A new stable local radial basis function approach for option pricing. Comput. Econ. **49**, 271–288 (2017)
4. Han, H., Wu, X.: A fast numerical method for the Black-Scholes equation of American options. SIAM J. Numer. Anal. **41**, 2081–2095 (2003)
5. Ikonen, S., Toivanen, J.: Operator splitting methods for American option pricing. Appl. Math. Lett. **17**, 809–814 (2004)
6. Jackwerth, J., Rubinstein, M.: Recovering stochastic processes from option prices. Working Paper, University of California, Berkeley, CA (2001)
7. Larguinho, M., Dias, J.C., Braumann, C.A.: On the computation of option prices and Greeks under the CEV model. Quant. Finan. **13**, 907–917 (2013)
8. Schmelzer, T., Trefethen, L.N.: Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integrals. Electron. T Numer. Ana. **29**, 1–18 (2007)
9. Tangman, D.Y., Gopaul, A., Bhuruth, M.: Exponential time integration and Chebychev discretisation schemes for fast pricing of options. Appl. Numer. Math. **58**, 1309–1319 (2008)

10. Thakoor, N.: Analytical shape functions and derivatives approximation formulas in local radial point interpolation methods with applications to financial option pricing problems. Comput. Math. Appl. **78**, 3770–3789 (2019)
11. Thakoor, N., Behera, D.K., Tangman, D.Y., Bhuruth, M.: A new Howard-Crandall-Douglas algorithm for the American option problem in computational finance. In: Behera, H., Nayak, J., Naik, B., Abraham, A. (eds.) Computational Intelligence in Data Mining, vol. 711, pp. 5–13. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8055-5_13
12. Thakoor, N., Tangman, D.Y., Bhuruth, M.: A new fourth-order scheme for option pricing under the CEV model. Appl. Math. Lett. **26**, 160–164 (2013)
13. Thakoor, N., Tangman, D.Y., Bhuruth, M.: Efficient and high accuracy pricing of barrier options under the CEV diffusion. J. Comput. Appl. Math. **259**, 182–193 (2014)
14. Thakoor, N., Tangman, D.Y., Bhuruth, M.: Fast valuation of CEV American options. Wilmott Mag. **2015**, 54–61 (2015)
15. Thakoor, N., Tangman, D.Y., Bhuruth, M.: RBF-FD schemes for option valuation under models with price-dependent and stochastic volatility. Eng. Anal. Boundary Elem. **92**, 207–217 (2018)
16. Wong, H.Y., Zhao, J.: An artificial boundary method for American option pricing under the CEV model. SIAM J. Numer. Anal. **46**, 2183–2209 (2008)