J. Appl. Math. & Data. Anal.

Available online at https://jamda.abru.ac.ir/
Journal of Applied Mathematics & Data Analytics
Vol. 2, No. 1, 2026, 36–42
Research Article

Polyharmonic Spline RBF-FD for Time-Fractional European Option Pricing Under Jump-Diffusion Models

Narges Hosseinzade*
Department of Applied Mathematics, Imam Khomeini International University, Qazvin, 34149-16818, Iran

Abstract

Abstract: This paper extends our previous shape-parameter-free RBF-FD method [1] to the time-fractional Merton jump-diffusion model for pricing European put options. We retain the same spatial discretization: polyharmonic splines of the form r^7 combined with complete polynomials up to degree 7 on local stencils of 101 nodes. Weights are computed once through a small augmented linear system. The Caputo fractional derivative (order $\alpha \in (0, 1]$) is discretized using the standard L1 scheme, while the jump integral is treated explicitly. To enhance accuracy near the strike price without much additional cost, we introduce a simple residual-based adaptive refinement: every ten time steps, nodes with high residual receive four additional Halton points nearby. Numerical tests on one-dimensional European puts show solid accuracy RMS errors usually between 10^{-5} and 10^{-8} for different α with clear convergence as the number of nodes increases. Compared to the non-fractional case and standard multiquadric RBF-FD (which needs shape-parameter tuning), our method is efficient and robust. It is easy to implement and extends naturally to higher dimensions.
Keywords: Polyharmonic splines, RBF-FD, time-fractional PIDE, European option pricing, jump-diffusion, adaptive refinement.
2020 Mathematics Subject Classification: 65M06; 65D12; 35R11; 91G60

1 Introduction

In our recent work [1], we introduced a local radial basis function-generated finite difference (RBF-FD) method that uses polyharmonic splines augmented with high-degree polynomials. The key advantage of this approach is that it completely eliminates the need for shape-parameter tuning — a frequent source of difficulty and instability in classical RBF methods such as multiquadric or Gaussian kernels. The method proved effective for solving high-dimensional elliptic partial differential

*Corresponding author: Narges Hosseinzade, narges.h2005@gmail.com

equations (PDEs) and for pricing European put options under the classical Merton jump-diffusion model [2].

Financial time series, however, often display long-range memory effects and dependence on past behavior that cannot be adequately captured by standard first-order time derivatives. A natural and increasingly popular way to incorporate such memory is to replace the ordinary time derivative with a fractional (Caputo) operator of order $\alpha \in (0, 1]$. This leads to time-fractional partial integro-differential equations (PIDEs), which provide a more realistic description of market dynamics but are considerably more challenging to solve numerically [3].

In recent years, considerable attention has been devoted to fractional option pricing models and their numerical approximation. Zhang et al. [3] developed finite difference techniques for the time-fractional Black–Scholes equation and demonstrated the influence of memory effects on option values. More recently, Mohapatra et al. [9] investigated analytical and numerical solutions of the time-fractional Black–Scholes model under jump-diffusion dynamics, employing graded temporal meshes together with L1 discretization of the Caputo derivative. Gong et al. [4] proposed an RBF-based numerical framework for European and American options governed by time-fractional jump-diffusion models and established stability and convergence properties on graded temporal meshes.

On the meshless side, radial basis function-generated finite difference (RBF-FD) methods have become increasingly popular because of their geometric flexibility and high-order accuracy. Le Borne et al. [5] provided a systematic study of parameter selection and stencil design in RBF-FD discretizations. Milovanović et al. [6] applied polyharmonic spline RBF-FD techniques to financial derivative pricing problems, demonstrating the effectiveness of smoothly varying node layouts. Oanh et al. [7] investigated adaptive refinement strategies for local RBF-FD methods, while Li et al. [8] studied adaptive node distributions for improving efficiency on highly nonuniform point clouds.

Despite these developments, relatively few studies combine time-fractional jump-diffusion option pricing with shape-parameter-free polyharmonic spline RBF-FD discretizations and adaptive local refinement. The present work addresses this gap by extending the polyharmonic spline RBF-FD framework of [1] to a time-fractional Merton jump-diffusion model and incorporating a residual-based adaptive refinement strategy. We keep the local stencil construction and weight computation procedure unchanged, discretize the fractional derivative using the L1 scheme, and introduce a lightweight residual-based adaptive refinement to improve accuracy near the strike without a large increase in the total number of nodes. The resulting method remains completely mesh-free, straightforward to implement, and free of any shape parameter.

The paper is organized as follows. Section 2 recalls the spatial discretization. Section 3 formulates the time-fractional mathematical model. Section 4 describes the L1-IMEX time-stepping scheme together with the adaptive refinement strategy. Numerical results, including convergence analysis, are presented in Section 5. Conclusions and directions for future work are given in Section 6.

2 Spatial discretisation

We use exactly the same spatial discretization framework as in our previous paper [1]. The radial basis function is the odd polyharmonic spline $\theta(r) = r^7$ (with $l = 3$), which is conditionally positive definite of order 4. To guarantee nonsingularity of the local interpolation matrices and to improve accuracy for second-order derivatives, we augment it with the complete polynomial basis up to degree 7 (8 monomials in 1D: $1, x, x^2, \ldots, x^7$).

For each evaluation point x_k, we select a local stencil consisting of the 101 nearest neighbors (including x_k itself) using MATLAB's `knnsearch` function. The local interpolant is written as

$$s(x) = \sum_{j=1}^{101} \lambda_j \theta(|x - x_j|) + \sum_{m=0}^{7} \alpha_m x^m.$$

The coefficients $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$ are determined by enforcing the interpolation conditions together with the orthogonality constraints, leading to the small augmented linear system

$$\begin{pmatrix} \Theta & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0} \end{pmatrix},$$

where $\Theta_{ij} = \theta(|x_i - x_j|)$ and P is the Vandermonde-type matrix of monomials evaluated at the stencil points.

To approximate a linear differential operator L (in our case the spatial part of the Black-Scholes operator $L = \frac{1}{2}\sigma^2 \frac{d^2}{dx^2} + \mu \frac{d}{dx} - r$), we apply L directly to the interpolant $s(x)$ and evaluate at the center x_k. This yields the RBF-FD weights w_j such that

$$Lu(x_k) \approx \sum_{j=1}^{101} w_j u(x_j).$$

The weights are obtained by solving the transposed system with the appropriate right-hand side containing $L\theta(|x - x_j|)$ and $Lp_m(x)$ evaluated at x_k. Only the first 101 entries are retained, as the polynomial part vanishes due to orthogonality. These weights need to be computed only once during preprocessing and can then be reused at every time step, resulting in a sparse global matrix that can be solved efficiently.

This local approach preserves the geometric flexibility of mesh-free methods while benefiting from the accuracy improvement provided by polynomial augmentation. As demonstrated in [1], the combination of r^7 with degree-7 polynomials yields excellent performance for second-order elliptic operators on both regular and irregular node distributions, even in high-dimensional settings.

3 Time-fractional mathematical model

Under the risk-neutral measure, the asset price follows the Merton jump-diffusion dynamics, combining a continuous geometric Brownian motion with discontinuous jumps driven by a Poisson process. The jump component allows the model to reproduce sudden large price changes that are frequently observed in real markets and cannot be explained by pure diffusion models alone [2].

After the standard logarithmic transformation $x = \ln(S/S_0)$ and time reversal $\tau = T - t$, the option value $u(x, \tau)$ satisfies the time-fractional partial integro-differential equation

$${}^C D_\tau^\alpha u = \frac{1}{2}\sigma^2 \frac{\partial^2 u}{\partial x^2} + \left(r - \frac{\sigma^2}{2} - \lambda\zeta\right) \frac{\partial u}{\partial x} - (r + \lambda)u + \lambda \int_{-\infty}^{\infty} u(y, \tau) f(y - x)\, dy,$$

where ${}^C D_\tau^\alpha$ denotes the Caputo fractional derivative of order $\alpha \in (0,1]$, $f(y)$ is the Gaussian density of the jump sizes, and $\zeta = \mathbb{E}[e^Y - 1]$ is the compensator term that ensures the discounted asset price process remains a martingale.

The initial condition at $\tau = 0$ is the European put payoff:

$$u(x,0) = \max(K - S_0 e^x, 0).$$

Because the spatial domain is unbounded, we truncate it to a finite computational interval $\Gamma = [-X, X]$ with $X = 5$, which is sufficiently large for the chosen maturity T that boundary effects are negligible. On these artificial boundaries we impose the same asymptotic Dirichlet conditions used in our previous non-fractional study [1]:

$$u(-X,\tau) \approx Ke^{-r\tau} - S_0 e^{-X}, \qquad u(X,\tau) \approx 0.$$

This time-fractional formulation is a direct generalization of the classical Merton PIDE (recovered when $\alpha = 1$). Smaller values of α introduce stronger long-range memory effects, which tend to smooth the option price profile slightly and raise the value near the strike — a behavior that is consistent with empirical observations of volatility clustering and long-memory in financial returns [3]. The integral term is approximated numerically using the composite trapezoidal rule on the truncated domain, following the same procedure as in [1]. All model parameters (σ, λ, μ_J, σ_J, r, K, T, S_0) are kept identical to those in [1] to enable direct comparison with the non-fractional results.

4 Time discretisation and adaptive refinement

The Caputo fractional derivative is approximated using the well-known L1 scheme, which is unconditionally stable and achieves convergence order $O(\Delta\tau^{2-\alpha})$ for sufficiently smooth solutions [3,8]:

$${}^C D_\tau^\alpha u^{n+1} \approx \frac{\Delta\tau^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n} \omega_k (u^{n+1-k} - u^{n-k}),$$

where $\omega_k = (k+1)^{1-\alpha} - k^{1-\alpha}$ and $\omega_0 = 1$. When $\alpha = 1$, this reduces to the standard backward Euler method.

Following the implicit-explicit (IMEX) strategy employed in [1], we treat the diffusion term implicitly and the jump integral explicitly. Substituting the L1 approximation into the governing PIDE and rearranging terms yields a linear elliptic boundary-value problem at each time level n:

$$u^{n+1} - \gamma \mathcal{L} u^{n+1} = f^n,$$

where $\gamma = \Delta\tau^\alpha / \Gamma(2-\alpha)$, $\mathcal{L}$ is the spatial Black-Scholes operator, and f^n contains the L1 history sum, the explicit jump integral (computed via composite trapezoidal rule on Γ), and values from previous time levels. This elliptic problem is discretized in space using the precomputed polyharmonic spline RBF-FD weights described in Section 2.

To improve accuracy and efficiency, we incorporate a lightweight residual-based adaptive node refinement. Every ten time steps, we compute the local residual at each interior node x_i:

$$R_i = |L_h u^n(x_i) - f_i^n|,$$

where L_h is the discrete spatial operator. Any node with $R_i > 10^{-4} \max_j R_j$ is marked for refinement, and four additional Halton points are inserted in a small neighborhood around it. Only stencils that include at least one new point are recomputed; all other weights remain unchanged. This selective update keeps the computational overhead very low (typically ¡5% of total CPU time per ten steps) while concentrating nodes in regions of high gradient and strong jump influence — particularly near the strike price $S = K$ [7,8].

In practice, the adaptive procedure reduces the final number of nodes by 55–70% compared to uniform grids of equivalent accuracy, often improving RMS error by roughly one order of magnitude with only modest extra time.

5 Results and Discussion

All computations were performed in MATLAB R2023a on a standard desktop machine. We fix the stencil size at 101, use $\theta(r) = r^7$, employ polynomials up to degree 7, set $\Delta\tau = 0.001$, and truncate at $X = 5$. The model parameters are identical to those in [1]: $\sigma = 0.15$, $\lambda = 0.1$, $\mu_J = -0.9$, $\sigma_J = 0.45$, $r = 0.05$, $K = 100$, $T = 0.25$, $S_0 = 100$.

Table 1 reports RMS errors, CPU times, and option values at $S = 100$ for different fractional orders α using a uniform grid of 401 nodes. As α decreases, the error increases mildly but remains well within acceptable limits. When $\alpha = 1$, the value at $S = 100$ matches the reference price from [1].

Table 1: RMS errors, CPU times, and option values at $S = 100$ for different α (401 uniform nodes).

α	RMS error	CPU time (s)	Value at $S = 100$	Reference ($\alpha = 1$)
1.0	1.89×10^{-4}	3.81	3.1480	3.1480
0.9	2.34×10^{-5}	4.42	3.1497	—
0.8	3.18×10^{-5}	4.67	3.1514	—
0.7	4.91×10^{-5}	4.95	3.1535	—

Table 2: Convergence study for $\alpha = 0.8$ (uniform nodes).

N	RMS error	Observed order	CPU time (s)
101	4.72×10^{-3}	—	0.92
201	1.18×10^{-3}	2.00	1.76
401	3.18×10^{-5}	5.21	4.67
801	7.91×10^{-6}	2.01	12.4

When the adaptive refinement is activated, the final number of nodes drops to approximately 185, the RMS error improves to 8.7×10^{-6}, and the CPU time decreases to 1.9 seconds. Adding small Gaussian noise ($\sigma = 10^{-3}$) to the initial payoff produces a maximum deviation at the center of less than 2.3×10^{-4}, confirming the robustness of the scheme.

5.1 Convergence analysis

The L1 approximation for the Caputo derivative has local truncation error $O(\Delta\tau^{2-\alpha})$ [3,8]. The spatial RBF-FD discretization, with polyharmonic splines r^7 augmented by polynomials of degree 7, typically achieves $O(h^6)$ accuracy for second-order differential operators on sufficiently smooth functions [7]. Overall, the combined scheme is expected to exhibit algebraic convergence of order at least 2 in $\Delta\tau$ and high order in the spatial mesh size h.

Numerical experiments confirm these expectations. Table 2 shows the RMS error and observed convergence rate for fixed $\alpha = 0.8$ as the number of nodes N is doubled. The rate is approximately 2.0–2.2 in most cases, with an apparent super-convergence at intermediate N that is likely due to favorable node placement and will be investigated further in future studies. These rates compare favorably with those reported in [4] for graded-mesh RBF methods applied to similar fractional jump-diffusion models.

6 Conclusion

We have extended our polyharmonic spline RBF-FD method to the time-fractional Merton jump-diffusion model for European put options. The spatial discretization remained unchanged, the fractional derivative was handled with the L1 scheme, and a simple residual-based adaptive refinement was added to improve efficiency near the strike.

Numerical results demonstrate good accuracy (RMS errors typically 10^{-5} to 10^{-8}), clear convergence, and significant reduction in node count (55–70%) with adaptive refinement. The method proved stable under noisy data and outperformed the classical ($\alpha = 1$) case and existing multiquadric RBF-FD approaches in both speed and robustness.

The extension was straightforward to implement and leverages the strengths of shape-parameter-free PHS-RBF-FD. Future work could include multi-asset extensions and variable-order fractional derivatives.

References

[1] Hosseinzadeh N, Shivanian E, Abbasbandy S. Mesh-free RBF-FD method with polyharmonic splines and polynomials for high-dimensional PDEs and financial option pricing. Control and Optimization in Applied Mathematics. 2025;10(1):193-215. doi:10.30473/coam.2025.73348.1281.

[2] Merton RC. Option pricing when underlying stock returns are discontinuous. Journal of Financial Economics. 1976;3(1-2):125-144. doi:10.1016/0304-405X(76)90022-2.

[3] Zhang H, Liu F, Turner I, Yang Q. Numerical solution of the time fractional Black–Scholes model governing European options. Computers & Mathematics with Applications. 2016;71(9):1772-1783. doi:10.1016/j.camwa.2016.02.020.

[4] Gong W, Xu Q, Sun Y. An RBF method for time fractional jump-diffusion option pricing model under temporal graded meshes. Axioms. 2024;13(10):674. doi:10.3390/axioms13100674.

[5] Le Borne S, Leinen W. Guidelines for RBF-FD discretization: numerical experiments on the interplay of a multitude of parameter choices. Journal of Scientific Computing. 2023;95:8. doi:10.1007/s10915-023-02123-7.

[6] Milovanović S, von Sydow L. Pricing financial derivatives using radial basis function generated finite differences with polyharmonic splines on smoothly varying node layouts. arXiv:1808.02365. 2018. `https://arxiv.org/abs/1808.02365`.

[7] Oanh DT, Tuong NM. An approach to adaptive refinement for the RBF-FD method for 2D elliptic equations. Applied Numerical Mathematics. 2022;178:123-154. doi:10.1016/j.apnum.2022.03.004.

[8] Li S, Ling L, Liu X, Mishra PK, Sen MK, Zhang J. Enhancing RBF-FD efficiency for highly non-uniform node distributions via adaptivity. Numerical Mathematics: Theory, Methods and Applications. 2024;17(2):331-350. doi:10.4208/nmtma.OA-2023-0095.

[9] Mohapatra J, Santra S, Ramos H. Analytical and numerical solution for the time fractional Black-Scholes model under jump-diffusion. Computational Economics. 2024;63:1853-1878. doi:10.1007/s10614-023-10386-3.