A parallel time stepping approach using meshfree approximations for pricing options with non-smooth payoffs

G. E. Fasshauer[*], A. Q. M. Khaliq[†] and D. A. Voss[‡]

Abstract

In this paper we consider a meshfree radial basis function approach for the valuation of pricing options with non-smooth payoffs. By taking advantage of parallel architecture, a strongly stable and highly accurate time stepping method is developed with computational complexity comparable to the implicit Euler method implemented concurrently on each processor. This, in collusion with the radial basis function approach, provides an efficient and reliable valuation of exotic options, such as American digital options.

1 Introduction

The radial basis function (RBF) collocation method uses global shape functions to produce meshfree approximate solutions of PDEs with exponential convergence (see, e.g., [3] and references therein). In recent years meshfree RBF approximation has been considered by a number of authors as a means of solving the Black-Scholes equations for European as well as American options ([4]–[8]). However, their application to options with non–smooth payoffs has not been investigated. In this paper we consider a meshfree RBF approach as spatial approximation for pricing American options with non–smooth payoffs.

Non–smooth payoff functions can cause inaccuracies for numerical schemes when financial contracts are priced, in particular, valuing catastrophe bonds which exhibit features of instability due to the discontinuity of the payoffs of the digital (binary) options around their threshold [12]. When solving in time, several methods are available. Explicit schemes are easy to implement but suffer from stability problems as noticed by Hon and Mao [6] for vanilla options. Some well known second order implicit schemes such as the Crank Nicolson method, are prone to spurious oscillations and, due to discontinuity in the payoff function (or its derivative), severely deteriorate their convergence properties ([9], [11], [13]) rendering them unreliable for pricing even vanilla options.

The implicit Euler method does not experience oscillatory behavior, but rather smoothes the effects of discontinuities in the initial data without affecting the nature of the problem. In spite of its strong stability properties, however, the implicit Euler

[*]Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA
[†]Department of Mathematics, Knox College, Galesburg, IL 61401, USA
[‡]Department of Mathematics, Western Illinois University, Macomb, IL 61455, USA

method is only first order accurate. To achieve second order accuracy the possibility of using a multistep method, such as BDF2, would seem appropriate due to its strong stability properties. However, Windcliff *et al.* [15] noted that multistep methods do not perform optimally for complex American options, such as shout options.

In this paper, we take advantage of parallel architecture to develop a strongly stable time stepping method of fourth order which consists of a linear combination of four implicit Euler–like solves on four concurrent processors thereby retaining the smoothing property of the standard implicit Euler method. This increase in order does not come at the expense of any increase in ill-conditioning of the matrix systems when combined with a meshfree RBF approach for the valuation of the American digital put option.

In the case of digital options, where the payoff is non–smooth, the American put option takes the form

$$\begin{aligned} \frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP &= 0, \quad S > E, \quad 0 \le t < T, \\ P(S,T) &= \begin{cases} 1 & S \le E \\ 0 & S > E \end{cases}, \\ \lim_{S\to\infty} P(S,t) &= 0, \\ P(S,t) &= 1, \quad 0 \le S \le E. \end{aligned} \tag{1}$$

The simple transformation $S = e^y$ changes the PDE in (1) to

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 U}{\partial y^2} + (r - \frac{1}{2}\sigma^2)\frac{\partial U}{\partial y} - rU = 0, \tag{2}$$

leaving the initial and boundary conditions unchanged.

The outline of this paper is as follows. In section 2 we consider the development of multiquadric basis functions for the valuation of options along with a time stepping procedure. In section 3 we perform numerical experiments and provide a comparative analysis of the approach. Concluding remarks are given in section 4.

2 Development of Methods

2.1 Meshfree Approximations

Given N distinct data points (centers), y_j, we interpolate the unknown function U by the following radial basis functions ϕ_j:

$$U(y,t) \approx \sum_{j=1}^{N} \alpha_j(t)\phi_j(y), \tag{3}$$

where α_j are unknown coefficients depending on time and $\phi_j(y) = \phi(|y - y_j|)$. We will use multiquadric (MQ) RBFs with shape parameter $c = 4d_{min}$, where d_{min} is the minimum distance between any two collocation points y_j:

$$\phi_j(y) = \sqrt{(y - y_j)^2 + c^2}. \tag{4}$$

This choice of shape parameter was suggested by Hon and Mao [6]. While no "optimal" value for c is known, more sophisticated strategies for choosing the shape parameter, such as the leave-one-out cross validation algorithm of Rippa [10], can be found in the literature.

Substituting (3) into (2) and collocating at N collocation points y_i (taken to coincide with the centers) yields the following system of N equations

$$\frac{\partial U(y_i,t)}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 U(y_i,t)}{\partial y^2} + (r - \frac{1}{2}\sigma^2)\frac{\partial U(y_i,t)}{\partial y} - rU(y_i,t) = 0, \tag{5}$$

where

$$\frac{\partial U(y_i,t)}{\partial t} = \sum_{j=1}^{N} \frac{d\alpha_j(t)}{dt}\phi_j(y_i), \tag{6}$$

$$\frac{\partial U(y_i,t)}{\partial y} = \sum_{j=1}^{N} \alpha_j(t)\frac{\partial \phi_j(y_i)}{\partial y}, \tag{7}$$

$$\frac{\partial^2 U(y_i,t)}{\partial y^2} = \sum_{j=1}^{N} \alpha_j(t)\frac{\partial^2 \phi_j(y_i)}{\partial y^2}, \tag{8}$$

in which

$$\frac{\partial \phi_j(y_i)}{\partial y} = \frac{(y_i - y_j)}{\sqrt{(y_i - y_j)^2 + c^2}}, \tag{9}$$

$$\frac{\partial^2 \phi_j(y_i)}{\partial y^2} = \frac{(y_i - y_j)}{\sqrt{(y_i - y_j)^2 + c^2}} - \frac{(y_i - y_j)^2}{((y_i - y_j)^2 + c^2)^{3/2}}. \tag{10}$$

In matrix form, (5) becomes

$$L\dot{\alpha} + \frac{1}{2}\sigma^2 L_{yy}\alpha + (r - \frac{1}{2}\sigma^2)L_y\alpha - rL\alpha = 0, \tag{11}$$

where α denotes the vector of unknown coefficients α_j, and L, L_y and L_{yy} are the $N \times N$ matrices with entries $\phi_j(y_i)$, $\frac{\partial \phi_j(y_i)}{\partial y}$, and $\frac{\partial^2 \phi_j(y_i)}{\partial y^2}$ given in (4), (9), and (10) respectively. The matrix L is invertible since the collocation points are assumed distinct and, consequently, equation (11) can be rewritten as linear homogeneous ODE system for the time dependent coefficients

$$\dot{\alpha} = -L^{-1}[\frac{1}{2}\sigma^2 L_{yy}\alpha + (r - \frac{1}{2}\sigma^2)L_y\alpha - rL\alpha] \equiv P\alpha, \tag{12}$$

where

$$P = rI - \frac{1}{2}\sigma^2 L^{-1}L_{yy} + (r - \frac{1}{2}\sigma^2)L^{-1}L_y. \tag{13}$$

2.2 Time Stepping Procedures

For a given time step τ, the exact solution of (12) satisfies the two term recurrence relation

$$\alpha(t-\tau) = e^{-\tau P}\alpha(t) \tag{14}$$

which forms the basis of various time-stepping methods through approximating the exponential function by rational functions. Diagonal $[m/m]$Padé approximations, $R_{m,m}(z) = \frac{P_m(z)}{Q_m(z)}$, may be used to approximate the exponential function in (14), where $z = \tau\lambda$ and λ an eigenvalue of P. Such algorithms involve the solution of linear systems wherein the coefficient matrix is a polynomial of degree m in τP. This exasperates the conditioning of the system thereby deteriorating the accuracy of the computed solution. To circumvent these effects, the diagonal Padé approximations can be decomposed into a partial fraction expansion to control the conditioning, however, this approach involves linear solves with complex arithmetic; a detailed discussion on this issue is given Voss *et al.* [13]. We consider rational approximations to e^z of the form

$$R(z) = \frac{N(z)}{D(z)} = \frac{1 + a_1 z + a_2 z^2 + a_3 z^3}{(1-b_1 z)(1-b_2 z)(1-b_3 z)(1-b_4 z)}, \tag{15}$$

where the coefficients a_i and b_i are real and $b_i > 0$. Approximation (15) will be of order p to the exponential function if $R(z) = e^z + C_{p+1} z^{p+1} + O(z^{p+2})$, where C_{p+1} denotes the error constant. It is well known that the maximum order attainable by $R(z)$ is four, and that the smallest error constant occurs in the case of repeated poles, $b_i = b$. A rational approximation to e^z is said to be *A–acceptable* if $|R(z)| < 1$ whenever $Re(z) < 0$ and *L–acceptable* if, in addition, $|R(z)| \to 0$ as $Re(z) \to -\infty$. We consider *L–acceptable* rational approximations to e^z as oscillations are prevalent in methods arising from the *A–acceptable* diagonal $[m/m]$Padé approximations, for example, the Crank–Nicolson method which is based on the $[1/1]$Padé approximation. The *L–acceptable*, fourth order $R(z)$ with minimal error constant, C_5^*, has $b_i = b \approx 0.572$ and $|C_5^*| \approx .02725$. Replacing the exponential function by such rational repeated pole approximations leads to inherently serial algorithms. Instead, we have constructed the *L–acceptable* rational approximation

$$R(z) \quad = \quad \frac{1 - \frac{54}{35}z + \frac{39}{140}z^2 + \frac{13}{42}z^3}{(1-z)(1-\frac{2}{5}z)(1-\frac{9}{14}z)(1-\frac{1}{2}z)} \tag{16}$$

with error constant, C_5, where $|C_5| = \frac{1}{28} \approx 0.0357 \approx |C_5^*|$. Since $R(z)$ possesses distinct real poles, it admits a partial fraction expansion of the form

$$R(z) = \sum_{n=1}^{4} \frac{w_n}{1 - b_n z}, \tag{17}$$

where the expansion coefficients are found in the usual way

$$w_n = \lim_{z \to \frac{1}{b_n}} (1 - b_n z) R(z),$$

Table 2.1: Definition of $R(z)$

n	1	2	3	4
b_n	1	$\frac{2}{5}$	$\frac{9}{14}$	$\frac{1}{2}$
w_n	$\frac{19}{45}$	$\frac{-2500}{153}$	$\frac{-2401}{255}$	$\frac{79}{3}$

and the ensuing algorithm can be implemented in a parallel computing environment. The expansion coefficients are given in Table 2.1. The recurrence relation (14) is then approximated by

$$\alpha(t-\tau) = R(-\tau P)\alpha(t). \tag{18}$$

Letting $U^n = L\alpha^n$ denote the approximation $U(y_i, T-n\tau)$, the resulting strongly stable, parallel time–stepping algorithm, denoted $L4$, based on recurrence relation (18) then becomes:

Parallel Algorithm ($L4$)

$$Initialize: \quad \alpha^0 = L^{-1}U^0$$

for $n = 1 \ldots M$

$$Solve \quad on \quad concurrent \quad processors: \quad (I + b_k\tau P)\mathbf{v}_k = \alpha^{n-1}, \quad k = 1, \ldots, 4$$

$$Compute: \quad \alpha^n = \sum_{k=1}^{4} w_k \mathbf{v}_k$$

boundary update

$$\begin{aligned} Compute: \quad U^n &= L\alpha^n \\ U^n(1) &= 1 \\ \alpha^n &= L^{-1}U^n \end{aligned}$$

end

The typically large and dense linear systems can be solved concurrently on four processors. Of course, further opportunity for parallelism exists in space, that is, in parallel numerical linear algebra routines for solving each system. The stability functions for the parallel $L4$ method and Crank Nicolson method are displayed in Figure 2.1. Notice that the stability function for the Crank Nicolson method approaches -1, while that of the strongly parallel $L4$ approaches 0 which naturally dampens the oscillations caused by high frequency data.

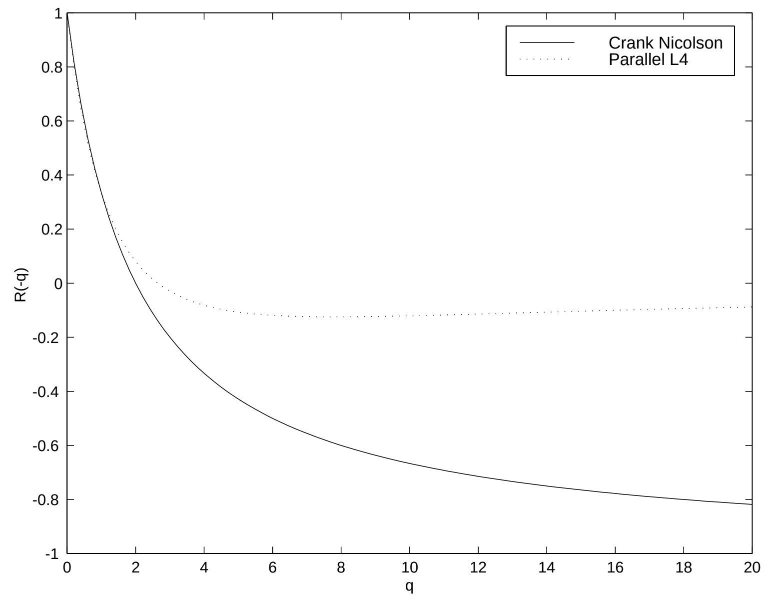

Figure 2.1: *Stability functions*

3 Numerical Experiments

The parallel algorithm developed in section 2 is implemented on the Black–Scholes PDE (1) with non–smooth payoff whose analytical solution is given in [14]. Table 3.1 contains a comparison of results between the exact solution and the RBF solutions for the Crank Nicolson (CN) and the parallel $L4$ method with $X = 1, r = 0.1, \sigma = 0.20, T = 1.0, S_{min} = 1, S_{max} = 2$ and $S \in \{1., 1.01, 1.02, 1.05, 1.1, 1.2, 1.4\}$. As the ratio of time step to space step ($\frac{\tau}{h}$) increases, the accuracy of the Crank Nicolson deteriorates while the $L4$ method produces a stable and accurate solution. This is confirmed in Figure 3.1 which depicts the oscillatory behavior of the Crank Nicolson method while the parallel $L4$ method returns a smooth and accurate solution.

Table 3.1: Root Mean Square Errors ($RMSE$) for Crank Nicolson (CN) and parallel $L4$ methods

N	M	$\frac{\tau}{h}$	RMSE	
			CN	L4
21	10	2.89	.3315	.0097
	15	1.92	.0158	.0079
	20	1.44	.0064	.0062
41	20	2.89	1737	.0210
	40	1.44	58.47	.0068
	80	0.72	.0062	.0062

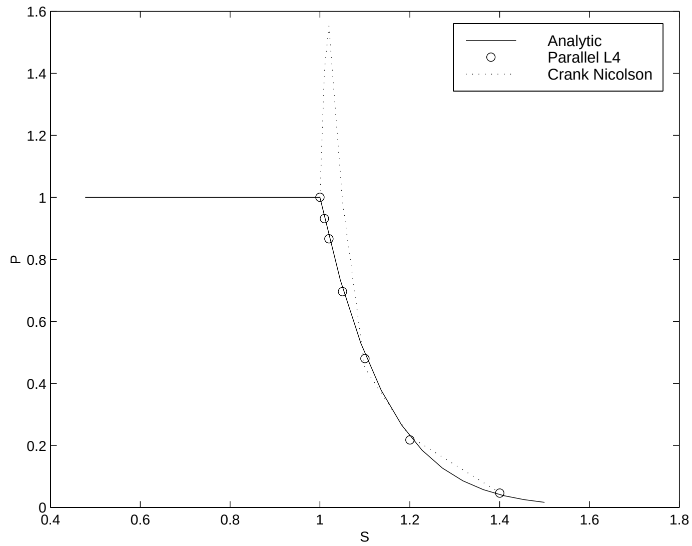

Figure 3.1: *Digital option:* $N = 21, M = 10, \sigma = 0.2$

Based on the performance of the parallel $L4$ method, we intend to consider it in combination with RBFs for application on multi–asset American options with more complicated payoff discontinuities, exotic options such as Asian and spread options [2], and numerical challenges associated with the valuation of these instruments. In addition, it is well known that the value of the multiquadric shape parameter balances the trade-off between achievable numerical accuracy and stability. Based on our experiments we propose to investigate the influence of the shape parameter in the presence of high volatility.

References

[1] F. Black and M. Scholes, The pricing of options and corporate liabilities, Political Economy 81 (1973), 637–655.

[2] R. Carmona and V. Durrleman, Pricing and hedging spread options, SIAM Review 45 (2003), 627–685.

[3] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa and G. Zammito, Exponential convergence and $H - c$ multiquadric collocation method for partial differential equations, Numer. Meth. PDE 19 (2003), 571–594.

[4] G. E. Fasshauer, A. Q. M. Khaliq, and D. A. Voss, Using meshfree approximations for multi–asset American option problems, submitted (2003).

[5] Y. C. Hon, A quasi-radial basis functions method for American options pricing, Comput. Math. Applic. 43 (2002) 513–524.

[6] Y. C. Hon and X. Z. Mao, A radial basis function method for solving options pricing models, Financial Engineering 8 (1999), 31–49.

[7] M. D. Marcozzi, S. Choi, and C. S. Chen, RBF and optimal stopping problems; an application to the pricing of vanilla options on one risky asset, *Boundary Element Technology* XIV, ed. C. S. Chen et al., Computational Mechanics Publications, (1999), 345–354.

[8] M. D. Marcozzi, S. Choi, and C. S. Chen, On the use of boundary conditions for variational formulations arising in financial mathematics, Appl. Math. Comput. 124 (2001), 197–214.

[9] D. M. Pooley, K. R. Vetzal, P. A. Forsyth, Convergence remedies for non–smooth payoffs in option pricing, J. Comp. Fin. 10 (2003), 25–40.

[10] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (1999), 193–210.

[11] D. Tavella and C. Randall, *Pricing Financial Instruments*, Wiley (New York) 2000.

[12] V. E. Vaugirard, Valuing catastrophe bonds by Monte Carlo simulations, Appl. Math. Fin. 10 (2003), 75–90.

[13] D. A. Voss, A. Q. M. Khaliq, S. H. K. Kazmi, and H. He, A fourth order L–stable method for the Black–Scholes model with barrier options, In M. L. Gavrilova, V. Kumar, and C. J. K. Tan (eds), *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg 2669, (2003) 199-207.

[14] P. Wilmott, S. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives: A Student Introduction*, Cambridge University Press, 1995.

[15] H. Windcliff, P. A. Forsyth, K. R. Vetzal, Shout options: A framework for pricing contracts which can be modified by the investor, J. Comp. Appl. Math. 134 (1999), 213–241.