

*Proceedings of the 13th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2013
24–27 June, 2013.*

Radial basis function methods in computational finance

Elisabeth Larsson¹, Sônia M. Gomes², Alfa Heryudono³ and Ali Safdari-Vaighani⁴

¹ *Department of Information Technology, Uppsala University, Sweden*

² *Department of Applied Mathematics, Universidade Estadual de Campinas, Brazil*

³ *Department of Mathematics, University of Massachusetts, Dartmouth, MA, USA*

⁴ *Department of Mathematics and Statistics, Allameh Tabatabai University, Tehran, Iran*

emails: elisabeth.larsson@it.uu.se, soniag@ime.unicamp.br,
aheryudono@umassd.edu, alisafdari87@gmail.com

Abstract

Radial basis function (RBF) based approximation methods for numerical solution of partial differential equations are interesting due to their potentially spectral accuracy and due to being meshfree. This could be especially beneficial for high dimensional problems, where meshing is non-trivial. In this work, we present different RBF approaches and evaluate them on a multi-asset option pricing problem. The conclusion is that the properties of the problem need to be taken into account in the solution method in order to have an approach that is viable for higher dimensions. Furthermore, we suggest to use an RBF based partition of unity approach in order to introduce locality and reduce the computational cost.

*Key words: radial basis function, option pricing, partition of unity, collocation
MSC 2000: 65M70*

1 Introduction

Radial basis function (RBF) based methods [4] have become quite popular for pricing of financial derivatives based on partial differential equation (PDE), or in the case of jump diffusion partial-integro differential equation (PIDE) formulations of the pricing problem. One of the main arguments is that RBF methods are easy to use in high-dimensions, i.e.,

for several underlying assets. The methods work only with scattered node points and do not require meshing. Furthermore, the basic mathematical formulation is the same in any number of dimensions. In an RBF method, the RBF approximation $s(\underline{x}, t)$ to the value $u(\underline{x}, t)$ of the financial derivative is typically of the form

$$s(\underline{x}, t) = \sum_{j=1}^N \lambda_j(t) \phi(\varepsilon \|\underline{x} - \underline{x}_j\|) \equiv \sum_{j=1}^N \lambda_j(t) \phi_j(\underline{x}), \quad (1)$$

where $\phi(r)$ is a (conditionally) positive definite RBF, ε is the shape parameter, which makes the RBF more flat as it goes to zero and more peaked as it goes to infinity, and x_j are scattered node points that act as the center points for the RBFs. The coefficients λ_j can be determined through collocation with equations and boundary conditions.

As can be seen from (1), the RBF approximation yields a continuous representation of the solution function. This allows explicit evaluation of derivatives of the approximation, which is an advantage in finance where the partial derivatives $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$, denoted by Δ and Γ , are needed for hedging purposes.

In the following sections, we will describe some different approaches, comment on their strengths and weaknesses, provide some relevant citations, and show numerical experiments to demonstrate the performance. Finally, we will reach to what we think is currently the most promising approach, radial basis function partition of unity methods, and present some preliminary results for these.

2 The model problems used for demonstrations

We will use the simplest possible option pricing problem to test the numerical approaches. We consider this to be a European basket call option, priced using the multi-dimensional Black-Scholes equation. Any added features like jump diffusion, stochastic volatility and exoticity may need special treatment by the numerical methods, but this is not an issue that we are pursuing in this paper.

The d -dimensional Black-Scholes equation for an option on d underlying assets is defined on \mathbb{R}_+^d . For computational purposes, we define a computational domain $\Omega \subset \mathbb{R}_+^d$. Furthermore, we define $\Gamma \subset \partial\Omega$ as the part of the boundary of the computational domain where we impose boundary conditions. After transformation of the time-variable [17] and scaling of the spatial variables as in [13], we can write the Black-Scholes equation as the following initial-boundary value problem

$$\frac{\partial u}{\partial t}(\underline{x}, t) = \mathcal{L}u(\underline{x}, t), \quad \underline{x} \in \Omega, \quad t > 0, \quad (2)$$

$$u(\underline{x}, t) = g(\underline{x}, t), \quad \underline{x} \in \Gamma, \quad t > 0, \quad (3)$$

$$u(\underline{x}, 0) = \Phi(\underline{x}), \quad \underline{x} \in \Omega, \quad (4)$$

where $u(\underline{x}, t)$ is the value of the option, $\underline{x} \in \mathbb{R}_+^d$ contains the scaled values of the d assets, and t is the time left to the exercise time T of the option. The spatial operator has the form

$$\mathcal{L}u(\underline{x}, t) = r \sum_{i=1}^d x_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^d [\sigma \sigma^T]_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} - ru,$$

where r is the risk free interest rate and σ is the volatility matrix. For our numerical examples, we use the contract function

$$\Phi(\underline{x}) = \max(0, \frac{1}{d} \sum_{i=1}^d x_i - K), \quad (5)$$

where, in our present case, the exercise price K is always equal to 1 due to scaling. The boundary conditions are linked to the contract function [19]. At the near-field boundary, consisting of the origin $\underline{x} = \underline{0}$, we use

$$g(\underline{x}, t) = 0, \quad (6)$$

and at the far-field boundary, here defined as the part of the boundary where $\frac{1}{d} \sum_{i=1}^d x_i \geq 4K$, we impose

$$g(\underline{x}, t) = \frac{1}{d} \sum_{i=1}^d x_i - K \exp(-rt). \quad (7)$$

3 Discretization in time and approximation in space

Let the time interval $[0, T]$ be divided into M steps of length $k^n = t^n - t^{n-1}$, $n = 1, \dots, M$, and let the approximate solution at the discrete times t^n be denoted by

$$v^n(\underline{x}) \approx u(\underline{x}, t^n).$$

In the majority of the numerical experiments we discretize the PDE problem (2–4) in time using the unconditionally stable, second-order accurate, implicit BDF-2 method [11, p. 401], resulting in

$$v^1(\underline{x}) - k^1 \mathcal{L}v^1(\underline{x}) = v^0(\underline{x}), \quad \underline{x} \in \Omega, \quad (8)$$

$$v^n(\underline{x}) - \beta_0^n \mathcal{L}v^n(\underline{x}) = \beta_1^n v^{n-1}(\underline{x}) - \beta_2^n v^{n-2}(\underline{x}), \quad \underline{x} \in \Omega, \quad n = 2, \dots, M, \quad (9)$$

$$v^n(\underline{x}) = g(\underline{x}, t^n), \quad \underline{x} \in \Gamma, \quad n = 1, \dots, M, \quad (10)$$

$$v^0(\underline{x}) = \Phi(\underline{x}), \quad \underline{x} \in \Omega, \quad (11)$$

The details of how we choose the coefficients β_i are described in [13]. For the approximation in space, we use (1) in its time discrete form, evaluated at the node points to get

$$v^n(\underline{x}_i) = \sum_{j=1}^n \lambda_j^n \phi_j(\underline{x}_i), \quad i = 1, \dots, N, \quad (12)$$

corresponding to the linear system

$$\underline{v}^n = A \underline{\lambda}^n, \quad (13)$$

where $\underline{v}^n = (v^n(\underline{x}_1), \dots, v^n(\underline{x}_N))^T$, $A_{ij} = \phi_j(\underline{x}_i)$, and $\underline{\lambda}^n = (\lambda_n(\underline{x}_1), \dots, \lambda_n(\underline{x}_N))^T$. In a similar fashion, we get

$$\mathcal{L} \underline{v}^n = B \underline{\lambda}^n, \quad (14)$$

where $B_{ij} = \mathcal{L} \phi_j(\underline{x}_i)$. Combining the two, we get

$$\mathcal{L} \underline{v}^n = B A^{-1} \underline{v}^n, \quad (15)$$

allowing us to work with nodal values as unknowns. Note that we can easily exchange the set of evaluation points $\{\underline{x}_i\}_{i=1}^N$ in the matrix B for some other set of points $\underline{x} \in \Omega$ to compute solution values or derivatives at arbitrary locations.

4 Numerical results for different RBF approaches

We use two different radial basis functions for the numerical experiments. The multiquadric RBF, which is conditionally positive definite, but nevertheless guarantees a non-singular interpolation matrix for distinct nodes and $\varepsilon > 0$,

$$\phi(r) = \sqrt{1 + \varepsilon^2 r^2},$$

and the Gaussian RBF, which is positive definite,

$$\phi(r) = \exp(-\varepsilon^2 r^2).$$

The scaled exercise price $K = 1$, and the exercise time used is $T = 1$ year. In the volatility matrix, we set $\sigma_{ii} = 0.3$ and $\sigma_{ij} = 0.05$, $i \neq j$. The risk free interest rate is set to $r = 0.05$. As computational domain we use $\Omega = \mathbb{R}_+^d \setminus \{\underline{x} \mid \frac{1}{d} \sum x_i > 4K\}$, which results in the interval $[0, 4]$ in 1-D, a triangle with corners in $(0, 0)$, $(0, 8)$, and $(8, 0)$ in 2-D, and higher order simplexes in more dimensions. This is possible because the RBF method is meshfree, and it leads significant savings in the computational cost compared with solving over a hypercube. This approach was used in [19] and [13]. In [13], we also showed that there is no loss of accuracy from this truncation of the domain.

Figure 1 shows examples of node layouts used in the numerical experiments in the 2-D case. The uniform and Chebyshev nodes can be generated for any number of dimensions

and are based on barycentric coordinates within the simplex [14]. In the Chebyshev case, the nodes are clustered in a Chebyshev fashion towards each boundary. The adapted nodes are more dense in the region of interest, and take the strike location into account.

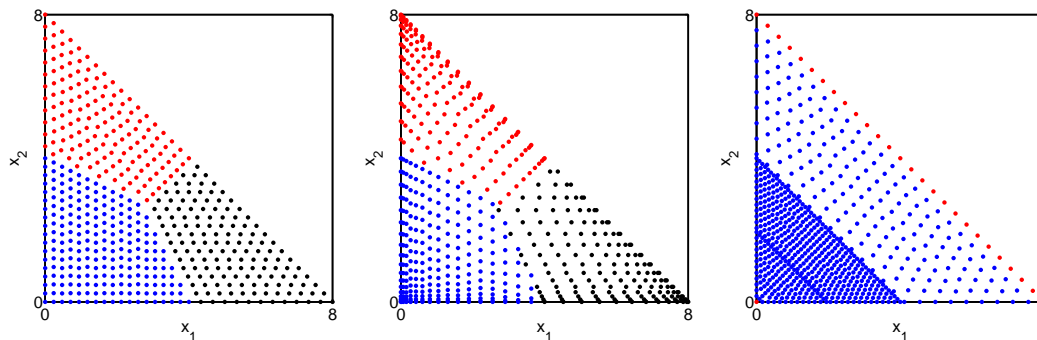


Figure 1: Examples of the *uniform* nodes, *Chebyshev* nodes, and *adapted* nodes that are used in our 2-D experiments.

In the following, the one-dimensional problem will be used as a starting point for quantitative investigation of different methods (due to the reasonable computational cost), the two-dimensional problem will be used for testing if the 1-D results carry over, and the potential for solving higher dimensional problems is discussed.

As asset prices are typically given with four or five digits of accuracy, we consider $\tau = 1 \cdot 10^{-4}$ to be a reasonable target accuracy for the solution of the option pricing problem. Of course, the desired accuracy significantly influences how large or how high-dimensional problems we can solve, so with a lower accuracy the projections become brighter. In the following subsections, we discuss different approaches in detail.

4.1 Global collocation using uniform node layouts

The most straightforward approach of an RBF method to option pricing is to use (1) directly on a set of uniformly distributed nodes. This was done for European and American options in one dimension by Hon et al. in [12, 23], and for one and two dimensions by Fasshauer et al. [6] and Marozzi et al. [16]. RBF methods have also been applied to other types of options and contracts such as a digital option [5], a currency option [2], and a credit default swap [10], as well as to problems with jump diffusion [1, 9, 22]. In all cases, the methods work well.

Figure 2 shows the results for different values of N as a function of ε . It should be noted that due to ill-conditioning that grows with increasing N and decreasing ε , the errors blow up and are not shown for the lower left corner of the figure. The target accuracy is reached the first time for $N = 46$ node points. The solutions are shown to the right and

the errors are about equally large at the boundaries and at the region of interest near the exercise price $K = 1$. It should also be noted that the results are sensitive to the placement of nodes near the strike discontinuity (see also [19]). Therefore, the number of node points have been chosen in the most favourable way.

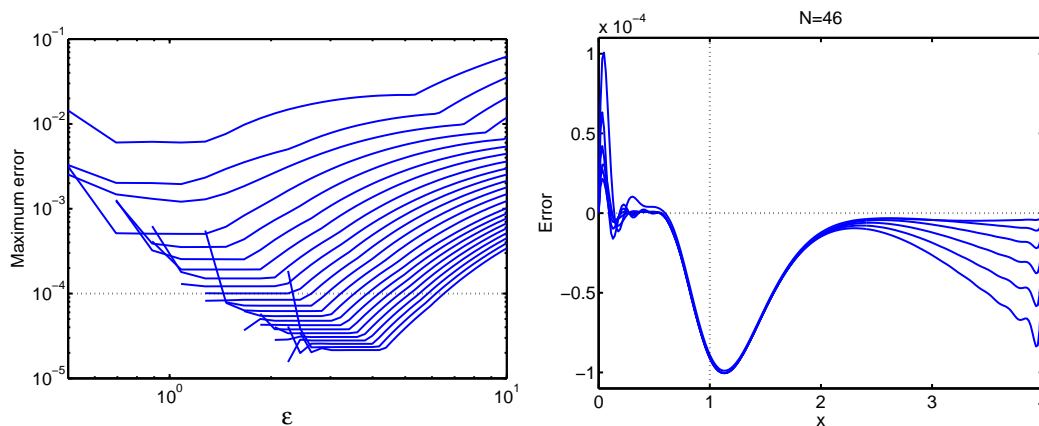


Figure 2: Left: The maximum error in the solution to the one-dimensional European option pricing problem as a function of ε for different values of N . From top to bottom, $N = 4j + 2$, $j = 2, \dots, 24$. The dotted line shows the target error. Right: The solution error as a function of the scaled asset price for different values of $\varepsilon \in (1.25, 2.25)$ for $N = 46$.

Now, consider the same problem in two dimensions. We can try to estimate how many points we are likely to need to achieve the same accuracy. It is reasonable to assume that we want the same node distance along the diagonal $x_1 = x_2$ in the two-dimensional case as we had along the interval in the one-dimensional case. This line is $\sqrt{2}$ times longer than the interval, leading to $\sqrt{2}N_{1D}$ nodes. We can use this as a measure of the number of points per dimension in the 2-D case. We can apply similar arguments in higher dimensions, leading to the special and general cases

$$N_{2D} \propto \frac{(\sqrt{2}N_{1D})^2}{2} = N_{1D}^2, \quad N_{dD} \propto \frac{(\sqrt{d}N_{1D})^d}{d!}, \quad (16)$$

where the factorial in the denominator is due to the ratio of the simplex to the hypercube. With 46 points in one dimension, this indicates that we need around 2100 points in two dimensions. The best solution we could come up with, without an extensive search of the parameter space, was for $\varepsilon = 1.5$ and $N = 2939$ with a maximum error $E = 5.2 \cdot 10^{-3}$. The error, displayed in Figure 5, is largest in the strike region. In the three-dimensional case, formula (16) indicates over 28 000 nodes. Since we need to solve a dense linear system of this size, this becomes very expensive, both in terms of computational cost and memory requirements. In four dimensions, we judge it to be unfeasible in practice.

4.2 Global collocation using adapted node layouts

As discussed in the previous subsection, the problem is sensitive to the placement of nodes near the strike region. Furthermore, the errors are large in this region, which is where we want to know the solution. Typically, options are traded with exercise prices in the vicinity of the current asset value.

By employing an adapted node layout, we aim to reduce the error in the region of interest, while possibly sacrificing accuracy in other parts of the domain. Therefore, we introduce a different error measure, the financial error [19], defined by

$$E_f = \max_{\underline{x} \in \Omega_K} |s(\underline{x}, T) - u(\underline{x}, T)|,$$

where $\Omega_K = \{\underline{x} \mid K - \frac{2}{3}K \leq \frac{1}{d} \sum_{i=1}^d x_i \leq K + \frac{2}{3}K\}$. We use the type of adapted node layout shown in Figure 1 and perform the same experiment as for the uniform nodes, but using the financial error measure. The results are shown in Figure 3. We can see that the errors in the strike region are much smaller than in the rest of the domain, and we reach the target accuracy already at $N = 17$ points. If we again use formula (16) we now need around 300 points in 2-D, 4000 points in 3-D, and 56 000 points in 4-D. This means that 3-D is definitely accessible, while 4-D might be stretching it a bit, but could be done with a lower target accuracy.

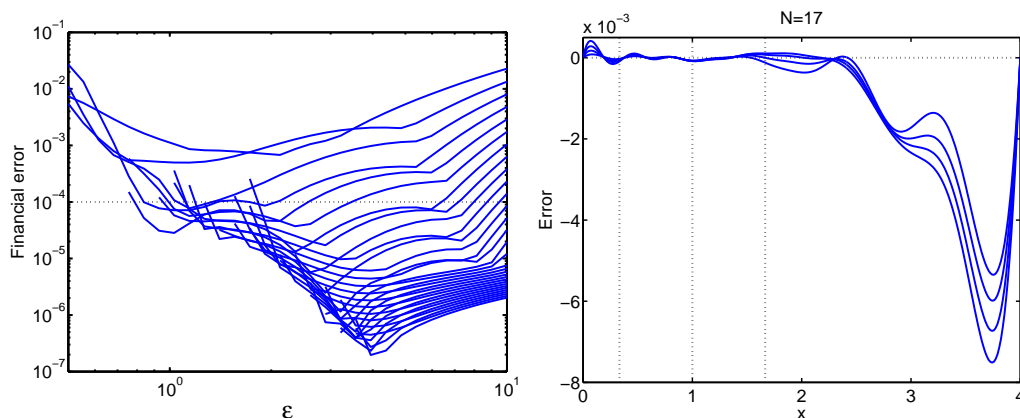


Figure 3: Left: The financial error for the adapted nodes as a function of ϵ for different values of N . From top to bottom, $N = 3j + 2$, $j = 3, \dots, 33$. The dotted line shows the target error. Right: The solution error as a function of the scaled asset price for different values of $\epsilon \in (1.03, 1.41)$ for $N = 17$.

Actual experiments in 2-D show that we can reach the target accuracy. However, because of the special node layout, we cannot hit the target exactly. At $N = 599$ points, we get a financial error $E_f = 5.1 \cdot 10^{-5}$. The result is shown in Figure 5.

The approach with adapted nodes is compared with the adaptive finite difference method from [18] in [19]. With our target tolerance, the adapted node RBF method is about 40 times faster in 1-D and 30 times faster in 2-D.

In [20], it is shown why using exponentially converging methods on uniform nodes must lead to exponential ill-conditioning. This issue can be overcome by clustering the nodes toward the boundaries, which was done successfully in [7]. However, for the option pricing problems, it results in reducing the errors at the boundaries while increasing them in the strike region, and is hence not an effective approach. Another example of an approach with node adaption that seems to work well is given in [1], where the adaptive residual subsampling method of [3] is used. There, the shape parameter is scaled individually for each RBF, proportionally to the inverse of the local node distance. This was suggested in [8], based on a heuristic exploration of optimal node locations and shape parameter values for some test problems. Here, we have employed this strategy for scaling the shape parameter in the adapted node approach. The results are shown in Figure 4. The error does become smaller outside of the strike region and the general behaviour of the error is somewhat improved, but there are no dramatic changes.

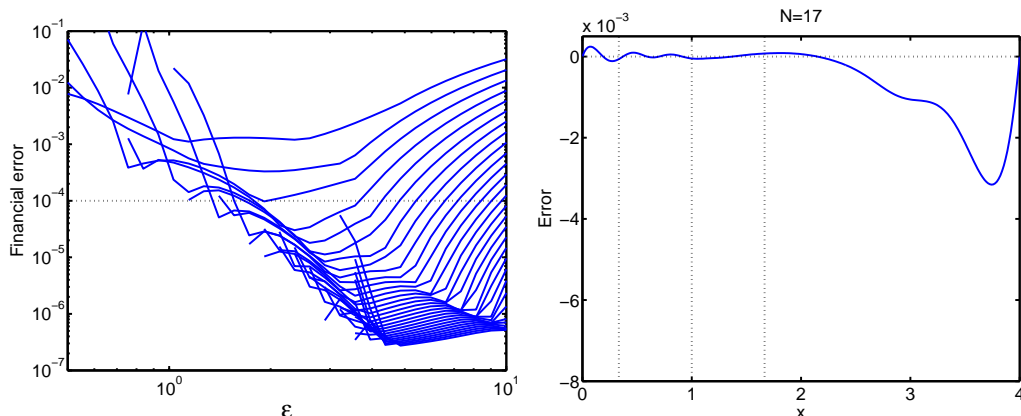


Figure 4: Results for adapted nodes and individually scaled shape parameters.

4.3 Least squares and multi-level approximations

Using a least squares approach, i.e., using more evaluation points than node points, leads to a better approximation of the non-smooth initial condition in terms of capturing the low frequencies compared with using pure collocation. By using uniform nodes and a least squares approximation we reach the target accuracy with $N = 26$ nodes. Solving the least squares problem is more expensive than solving the collocation system, but in [14], we show

that a least squares method is more effective than collocation in terms of the total work for a given accuracy.

An issue regarding the solutions to the option pricing problems that we have not mentioned so far is the transition from non-smooth to smooth. The non-smooth initial condition is best approximated using a large shape parameter, but the smooth solution at the final time fares better with a small shape parameter. The solution we propose in [14] is to use a multilevel approach with different shape parameters at different levels. This results in a method that is quite robust with respect to the choice of method parameters, that has a quite uniform error distribution and has a comparatively small error over the whole time interval.

Figure 5 shows how the least squares multilevel method compares with the other approaches. The number of nodes at the fine level (which determines the computational cost) is comparable to the adapted case, so the potential for solving higher dimensional problems is the same. However, the new method is more robust and the overall error behaviour is better. Therefore, we consider this to be the most promising approach so far.

5 The RBF partition of unity method

The main obstacle to using the different versions of global RBF methods is computational cost. An attractive compromise between high order and locality is offered by RBF based partition of unity (RBF-PU) methods. We have developed a method of this type in the manuscript [15]. This first paper deals with time-independent PDEs, and we are able to show theoretical results of the types below for the RBF-PU approximant.

$$\|s(\underline{x}) - u(\underline{x})\|_{W_\infty^2(\Omega)} \leq C \max_j C_j \rho_j^{m - \frac{d}{2} - \alpha} \|u\|_{\mathcal{N}(\Omega_j)}, \quad (17)$$

$$\|s(\underline{x}) - u(\underline{x})\|_{W_\infty^2(\Omega)} \leq C e^{\gamma \log(h)/\sqrt{h}} \max_j \|u\|_{\mathcal{N}(\Omega_j)}, \quad (18)$$

where Ω_j are the partitions that cover Ω , ρ_j is the radius of the partition Ω_j , h is the local node distance, and α is the degree of the PDE operator. An example of adapted partitions and nodes is shown in Figure 5. The meaning of the two estimates is the following

- (i) If we fix the number of nodes/partition, we get algebraic convergence in ρ .
- (ii) If we fix the partitions, we get spectral convergence in the local node distance.

A numerical demonstration of the theoretical results is given in Figure 6. Note that the target accuracy can be reached without using RBF-QR [7]. This is relevant for higher dimensions since RBF-QR is currently only available in up to three dimensions. The system matrix of the RBF-PU method is sparse, which allows us to solve very large systems of equations. We are currently working on a parallel iterative solver for these systems.

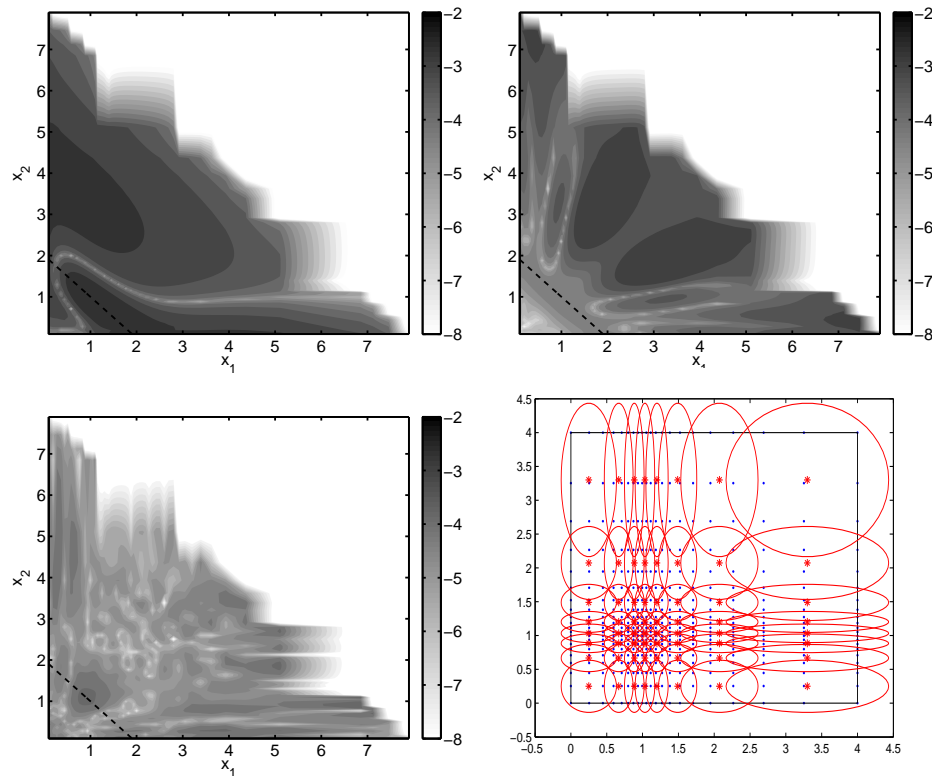


Figure 5: Top left: Collocation on $N = 2939$ uniform nodes for $\varepsilon = 1.5$. The dashed line indicates the location of the strike discontinuity. Top right: Collocation with $N = 599$ adapted nodes. Bottom left: The least squares multilevel method with $N_f = 592$ nodes at the fine level and $N_c = 96$ at the coarse level for $\varepsilon_f = 2$ and $\varepsilon_c = 0.1$. Bottom right: Partitions and nodes for the RBF-PU method.

In [21] the RBF-PU method is applied to a convection-diffusion problem and an American option pricing problem with promising results. The node and partition layout used for the American option pricing problem is shown in Figure 5.

References

- [1] R. T. L. CHAN, *Numerical analysis of American and European options under Lévy processes by meshless methods*, PhD thesis, Birckbeck, University of London, 2011.
- [2] S. CHOI AND M. D. MARCOZZI, *A numerical approach to American currency option valuation*, *The Journal of Derivatives*, 9 (2001), pp. 19–29.

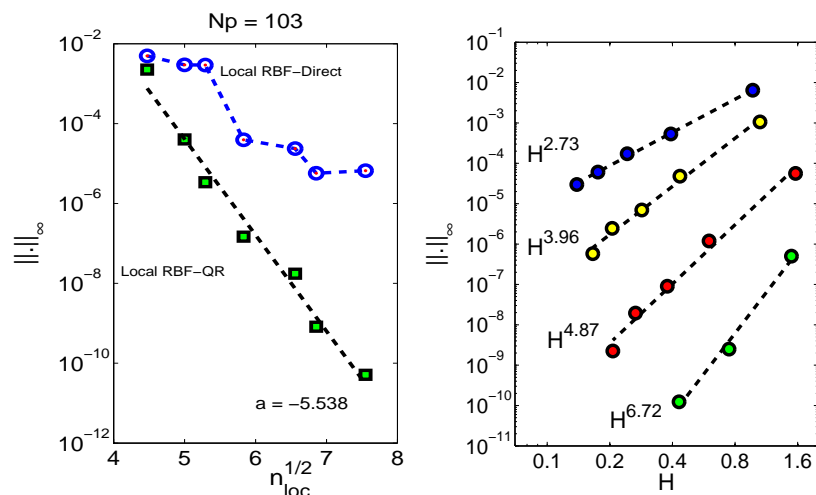


Figure 6: Left: Spectral convergence in the local node distance $h \propto \sqrt{n_{loc}}$. Right: Algebraic convergence in the partition size $H \propto \rho_j$ for increasing numbers of local nodes.

- [3] T. A. DRISCOLL AND A. R. H. HERYUDONO, *Adaptive residual subsampling methods for radial basis function interpolation and collocation problems*, *Comput. Math. Appl.*, 53 (2007), pp. 927–939.
- [4] G. E. FASSHAUER, *Meshfree approximation methods with MATLAB*, vol. 6 of *Interdisciplinary Mathematical Sciences*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.
- [5] G. E. FASSHAUER, A. Q. M. KHALIQ, AND D. A. VOSS, *A parallel time stepping approach using meshfree approximations for pricing options with non-smooth payoffs*, in *Third World Congress of the Bachelier Finance Society*, Chicago, IL, USA., July 2004.
- [6] ———, *Using meshfree approximation for multi-asset american option problems*, *J. of Chin. Inst. Eng.*, 27 (2004), pp. 563–571.
- [7] B. FORNBERG, E. LARSSON, AND N. FLYER, *Stable computations with Gaussian radial basis functions*, *SIAM J. Sci. Comput.*, 33 (2011), pp. 869–892.
- [8] B. FORNBERG AND J. ZUEV, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, *Comput. Math. Appl.*, 54 (2007), pp. 379–398.
- [9] A. GOLBABAI, D. AHMADIAN, AND M. MILEV, *Radial basis functions with application to finance: American put option under jump diffusion*, *Mathematical and Computer Modelling*, 55 (2012), pp. 1354–1362.

- [10] A. GUARIN, X. LIU, AND W. L. NG, *Enhancing credit default swap valuation with meshfree methods*, European Journal of Operational Research, 214 (2011), pp. 805–813.
- [11] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving ordinary differential equations. I*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1993. Nonstiff problems.
- [12] Y. C. HON AND X.-Z. MAO, *A radial basis function method for solving options pricing model*, Journal of Financial Engineering, 8 (1999), pp. 1–24.
- [13] E. LARSSON, K. ÅHLANDER, AND A. HALL, *Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform*, J. Comput. Appl. Math., 222 (2008), pp. 175–192.
- [14] E. LARSSON AND S. M. GOMES, *A least squares multi-level radial basis method with applications in finance*. manuscript in preparation, 2013.
- [15] E. LARSSON AND A. HERYUDONO, *A partition of unity radial basis function collocation method for partial differential equations*. manuscript in preparation, 2013.
- [16] 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), pp. 197–214.
- [17] J. PERSSON AND L. VON SYDOW, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci., 10 (2007), pp. 173–183.
- [18] J. PERSSON AND L. VON SYDOW, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci., 10 (2007), pp. 173–183.
- [19] U. PETTERSSON, E. LARSSON, G. MARCUSSON, AND J. PERSSON, *Improved radial basis function methods for multi-dimensional option pricing*, J. Comput. Appl. Math., 222 (2008), pp. 82–93.
- [20] R. B. PLATTE, L. N. TREFETHEN, AND A. B. J. KUIJLAARS, *Impossibility of fast stable approximation of analytic functions from equispaced samples*, SIAM Rev., 53 (2011), pp. 308–318.
- [21] A. SAFDARI-VAIGHANI, A. HERYUDONO, AND E. LARSSON, *A radial basis function partition of unity collocation method for convection diffusion equations*. manuscript in preparation, 2013.
- [22] A. A. E. F. SAIB, D. Y. TANGMAN, AND M. BHURUTH, *A new radial basis functions method for pricing American options under Merton's jump-diffusion model*, Int. J. Comput. Math., 89 (2012), pp. 1164–1185.
- [23] Z. WU AND Y. C. HON, *Convergence error estimate in solving free boundary diffusion problem by radial basis functions method*, Engng. Anal. with Bound. Elem., 27 (2003), pp. 73–79.