# A radial basis function partition of unity collocation method for convection-diffusion equations ⋆

A. Safdari-Vaighani [a], A. Heryudono [b], E. Larsson [c]

[a] *Department of Mathematics and Statistics, Allameh Tabataba'i University, Tehran, Iran.*

[b] *Department of Mathematics, University of Massachusetts Dartmouth, Dartmouth, Massachusetts, USA.*

[c] *Department of Information Technology, Uppsala University, Uppsala, Sweden.*

**Abstract**

Numerical solution of multi-dimensional PDEs is a challenging problem with respect to computational cost and memory requirements, as well as regarding representation of realistic geometries and adaption to solution features. Meshfree methods such as global radial basis function approximation have been successfully applied to several types of problems. However, due to the dense linear systems that need to be solved, the computational cost grows rapidly with dimension. In this paper, we instead propose to use a locally supported RBF collocation method based on a partition of unity approach to numerically solve time-dependent PDEs. We investigate the stability and accuracy of the method for convection-diffusion problems in two space dimensions as well as for an American option pricing problem. The numerical experiments show that we can achieve both spectral and high-order algebraic convergence for convection-diffusion problems, and that we can reduce the computational cost for the option pricing problem by adapting the node layout to the problem characteristics.

*Key words:* collocation method; meshless method; radial basis function; partition of unity; PUM; convection-diffusion equation; American option.

## 1 Introduction

Numerical option pricing is steadily gaining in popularity due to the increasing interest in and demand for complex financial derivatives. One example is financial contracts based on several underlying assets. Such option prices can be modeled by higher dimensional generalizations of the original Black–Scholes equation [2,13]. Fasshauer et al. [3] and Pettersson et al. [20] have shown that mesh-free methods based on radial basis functions (RBFs) may reduce the computational efforts significantly compared with finite difference methods for problems in one and two assets. Therefore, RBF methods are promising candidates for solving high-dimensional financial problems efficiently. Option pricing using RBFs was explored in one dimension for European and American options by Hon et al. [27,8]. Pettersson et al. [20] presented a RBF based method for multi-dimensional European options, and American options in both one and two dimensions are investigated by Fasshauer et al. [3]. All of these papers employ global RBF collocation methods.

Partition of unity (PU) schemes have been used for the construction of interpolants for a long time. This type of approach was taken by Shepard [23], McLain [16], and Franke and Nielson [7]. PU schemes were combined with RBFs for interpolation problems in [26] and [4]. In the last two decades, the partition of unity method (PUM) has gained importance for the construction of approximations to the solutions of partial differential equations (PDEs) due to the early work of Babuška and Melenk [1]. The main advantages of the PUMs are the flexibility in choosing ansatz spaces and their good approximation properties. The framework of the PUM allows different ways to adapt the local approximation to the solution. The size of the partitions can be changed (*h*-version) and the dimension and character of the function space for the local approximation can

⋆ The work of the second author was supported in part by grants from the AFOSR FA-9550-12-1-0224 and the NSF DMS 1318427.

*Email addresses:* a_safdari@iust.ac.ir (A. Safdari-Vaighani), aheryudono@umassd.edu (A. Heryudono), elisabeth.larsson@it.uu.se (E. Larsson).

be varied ($p$-version). Recently, Larsson and Heryudono [15] introduced a collocation PUM with local RBF approximations for numerical solution of time-independent PDEs. The RBF–PUM formulation has high (spectral) approximation order locally and generates sparse coefficient matrices, and is hence well suited for multi-dimensional problems [15].

In this paper, we introduce the RBF–PUM as an alternative method for numerical solution of time-dependent PDEs. We investigate the capability of the method for the two dimensional convection-diffusion equation, and as a second example, we consider the American multi-asset put option pricing problem.

## 2 Radial basis function collocation schemes

RBF methods are meshfree and work with data given at scattered node points. Given $N$ distinct points $x_1,\dots,x_N \in \mathbb{R}^d$ and corresponding scalar function values $u(x_1),\dots,u(x_N)$, the standard RBF interpolation problem is to find an interpolant of the form

$$s(x) = \sum_{j=1}^{N} \lambda_j \phi(\|x-x_j\|), \tag{2.1}$$

where $\|\cdot\|$ is the Euclidean norm, $\lambda_j \in \mathbb{R}$ for $j = 1,\dots,N$, and $\phi$ is a real-valued function such as the inverse multiquadric $\phi(r) = \frac{1}{\sqrt{\varepsilon^2 r^2+1}}$. The coefficients $\lambda_1,\dots,\lambda_N$ are determined by enforcing the conditions $s(x_i) = u(x_i)$, $i = 1,\dots,N$. Imposing these conditions leads to a symmetric linear system of equations

$$A\underline{\lambda} = \underline{u}, \tag{2.2}$$

where $A_{ij} = \phi(\|x_i - x_j\|),\ i,j = 1,\dots,N$, $\underline{u} = [u(x_1)\dots u(x_N)]^T$, and $\underline{\lambda} = (\lambda_1\dots\lambda_N)^T$. When $\underline{\lambda}$ is known, we can with this notation evaluate the RBF interpolant at a point $x$ as

$$s(x) = \bar{\phi}(x)\underline{\lambda}, \tag{2.3}$$

where $\bar{\phi}(x) = (\phi(\|x-x_1\|),\dots,\phi(\|x-x_N\|))$.

For the error analysis and also for the implementation of boundary conditions discussed later, it is preferable to express the interpolant in Lagrange form, i.e., using cardinal basis functions. The cardinal basis functions, $\psi_j(x)$, $j = 1,\dots,N$, have the property

$$\psi_j(x_i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \qquad j = 1,\dots,N, \tag{2.4}$$

leading to the alternative formulation for the interpolant

$$s(x) = \bar{\psi}(x)\underline{u}, \tag{2.5}$$

where $\bar{\psi}(x) = (\psi_1(x),\dots,\psi_N(x))$. Combining (2.3), (2.5), and (2.2) leads to the following relation between the cardinal basis and the original radial basis:

$$s(x) = \bar{\psi}(x)\underline{u} = \bar{\phi}(x)\underline{\lambda} = \bar{\phi}(x)A^{-1}\underline{u} \quad \Rightarrow \quad \bar{\psi}(x) = \bar{\phi}A^{-1}. \tag{2.6}$$

This transformation is valid whenever $A$ is non-singular. This holds for distinct node points $x_1,\dots,x_N$ and commonly used RBFs such as Gaussians, inverse multiquadrics and multiquadrics.

For a linear operator $\mathscr{L}$, we have

$$\mathscr{L}s(x) = \sum_{j=1}^{N} \mathscr{L}\psi_j(x) u(x_j). \tag{2.7}$$

To evaluate $\mathscr{L}s(x)$ at the node points, i.e., to evaluate $\underline{s}_{\mathscr{L}} = (\mathscr{L}s(x_1),\dots,\mathscr{L}s(x_N))^T$, we need the differentiation matrix $\Psi_{\mathscr{L}} = \{\mathscr{L}\psi_j(x_i)\}_{i,j=1,\dots,N}$. Using relation (2.6), this leads to

$$\underline{s}_{\mathscr{L}} = \Psi_{\mathscr{L}}\underline{u} = \Phi_{\mathscr{L}}A^{-1}\underline{u}, \tag{2.8}$$

where $\Phi_{\mathscr{L}} = \{\mathscr{L}\phi(\|x_i - x_j\|)\}_{i,j=1,\ldots,N}$.

When the Lagrangian form of the RBF interpolation method is used in the context of solving a time-dependent PDE problem, the solution $u(x,t)$ is approximated by

$$s(x,t) = \sum_{j=1}^{N} \psi_j(x) u_j(t), \tag{2.9}$$

where $u_j(t) \approx u(x_j,t)$ are the unknown functions to determine.

## 3 The radial basis function based PUM

This section defines the RBF–PUM collocation method for time-dependent PDEs in terms of its weight functions and local RBF approximations.

### *3.1 The partition of unity weight functions*

Let $\Omega \subset \mathbb{R}^d$ be an open set, and let $\{\Omega_i\}_{i=1}^M$ be an open cover of $\Omega$ satisfying a pointwise overlap condition and that

$$\forall x \in \Omega \quad I(x) = \{j | x \in \Omega_j\}, \quad \text{card}(I(x)) \leq K,$$

where $K$ is a constant that does not depend on $M$. In the RBF–PUM, the global approximation function $s(x)$ in $\Omega$ to the solution function $u(x)$ is constructed as follow

$$s(x) = \sum_{j=1}^{M} w_j(x) s_j(x), \tag{3.1}$$

where $s_j$ is an RBF approximation of $u(x)$ on patch $\Omega_j$ and $w_j : \Omega_j \to \mathbb{R}$ are compactly supported, nonnegative weight functions subordinate to the cover. The partition of unity weight functions $w_j$, which also occur under the name *shape functions*, are constructed using Shepard's method as follows

$$w_j(x) = \frac{\varphi_j(x)}{\sum_{k \in I(x)} \varphi_k(x)}, \quad j = 1, \ldots, M, \tag{3.2}$$

where $\varphi_j(x)$ are compactly supported functions with support on $\Omega_j$. Here, we select compactly supported Wendland functions [25] such as

$$\varphi(r) = \begin{cases} (1-r)^4(4r+1) & \text{if } 0 \leq r \leq 1, \\ 0 & \text{if } r > 1, \end{cases} \tag{3.3}$$

for the construction of the weight functions. Non-negativity and compact support on radially symmetric patches $\Omega_j$ are guaranteed if the weight functions are constructed using

$$\varphi_j(x) = \varphi\left(\frac{\|x - \mathbf{x}_j\|}{r_j}\right), \quad j = 1, \ldots, M, \tag{3.4}$$

where $r_j$ is the radius of the patch $\Omega_j$, and $\mathbf{x}_j$ is its center point. It follows from (3.2) that the weight functions $w_j(x)$ satisfy the partition of unity property

$$\sum_{j \in I(x)} w_j(x) = 1. \tag{3.5}$$

Moreover, the equations (3.3)-(3.4) show that $w_j(x) = 0, \ \forall j \notin I(x)$. Therefore, equation (3.1) can be rewritten as

$$s(x) = \sum_{j \in I(x)} w_j(x) s_j(x). \tag{3.6}$$

If the functions $s_j(x),\ j=1,\ldots,M$ from equation (3.6) are local interpolants with $s_j(x_i)=u(x_i)$ for each node point $x_i\in\Omega_j$, then the global PU approximant inherits the interpolation property of the local interpolants, i.e.

$$s(x_i)=\sum_{j\in I(x_i)} w_j(x_i)s_j(x_i)=u(x_i)\sum_{j\in I(x_i)} w_j(x_i)=u(x_i). \tag{3.7}$$

The patches can be of any shape: square, circular, etc. The common requirement for all shapes of patches is that they cover the domain and the boundary. In this paper, circle and oval shapes of the patches will be investigated. In the case of oval patches, the functions used for generating the weight functions are adjusted accordingly, since the support and the shape of the patch must match each other.

When we use these types of patches, the overlap between patches can be regulated, and covering ensured, by adjusting the radius of the patches. Flexibility in selection of the radius of the patches is another advantage of the local properties of the PUM. Figures 4 and 12 demonstrate the discretization of a square domain with circle and oval patches.

### *3.2 The RBF–PUM for time-dependent problems*

In the RBF–PUM, the solution $u(x,t)$ for a time-dependent problem is approximated by

$$s(x,t)=\sum_{j=1}^{M} w_j(x)s_j(x,t), \tag{3.8}$$

where $s_j(x,t)$ is an RBF approximant of the type (2.9) on $\Omega_j$, i.e.

$$s_j(x,t)=\sum_{k\in J(\Omega_j)} \psi_k(x)u_k(t), \tag{3.9}$$

where $J(\Omega_j)=\{k|x_k\in\Omega_j\}$ is the set of node points in $\Omega_j$.

From the equations (3.8) and (3.9), we can express the global approximant as

$$\begin{aligned} s(x,t)&=\sum_{j=1}^{M} w_j(x)\sum_{k\in J(\Omega_j)}\psi_k(x)u_k(t)=\sum_{j\in I(x)} w_j(x)\sum_{k\in J(\Omega_j)}\psi_k(x)u_k(t)\\ &=\sum_{j\in I(x)}\sum_{k\in J(\Omega_j)}\Big(w_j(x)\psi_k(x)\Big)u_k(t), \end{aligned} \tag{3.10}$$

Note that by interpolating the initial condition we get $s(x_k,0)=u(x_k,0)$ for all $k$, but $s(x_k,t)\approx u(x_k,t)$ for $t>0$.

### *3.3 Differentiating the RBF–PU approximant*

In order to use the RBF–PU approximation (3.10) for a PDE problem, we need to compute the effect of applying a spatial differential operator $\mathscr{L}$ at the interior node points. Let $\alpha$ and $\beta$ be multi-indices and adopt common rules for multi-index notation. Then a derivative term of order $\alpha$ in the differential operator can be applied in two steps. First we use the properties of the local approximations in each patch to compute $\frac{\partial^{|\beta|}\psi_k}{\partial x^\beta}(x_i)$ and then we compute the values of $\frac{\partial^{|\alpha-\beta|}\omega_j}{\partial x^{\alpha-\beta}}(x_i)$ for all $\beta\le\alpha$. The derivative applied to the global approximation (3.10) becomes

$$\begin{aligned} \frac{\partial^{|\alpha|}}{\partial x^\alpha}s(x_i,t)&=\sum_{j\in I(x_i)}\sum_{k\in J(\Omega_j)}\frac{\partial^{|\alpha|}}{\partial x^\alpha}[w_j(x)\psi_k(x)]_{x=x_i}u_k(t)\\ &=\sum_{j\in I(x_i)}\sum_{k\in J(\Omega_j)}\left[\sum_{\beta\le\alpha}\binom{\alpha}{\beta}\frac{\partial^{|\alpha-\beta|}\omega_j}{\partial x^{\alpha-\beta}}(x_i)\frac{\partial^{|\beta|}\psi_k}{\partial x^\beta}(x_i)\right]u_k(t),\quad 1\le i\le N_I, \end{aligned} \tag{3.11}$$

where $N_I$ is the total number of interior node points in $\Omega$. Fixing $i$ and $k$ in equation (3.11) gives us the $ik$-element of the global differentiation matrix corresponding to the $\alpha$-derivative. For composite linear operators, we just sum up the contributions from each term. We denote the global differentiation matrix under operator $\mathscr{L}$ by $W_{\mathscr{L}}$.

## 4 The unsteady convection-diffusion equation

Consider an unsteady convection-diffusion equation of the form

$$\frac{\partial u(x,t)}{\partial t} = \kappa \Delta u(x,t) + v \cdot \nabla u(x,t) \equiv \mathscr{L} u(x,t), \quad x \in \Omega \subset \mathbb{R}^d,\ t > 0, \tag{4.1}$$

where $\Delta$ and $\nabla$ denote the Laplacian and the gradient operator, respectively, $\kappa$ is the diffusion coefficient, $v$ is a constant velocity vector, and $u(x,t)$ may represent concentration or temperature for mass or heat transfer, respectively. This equation also serves as a simplified model problem for the Black–Scholes equation and other equations in financial mathematics. The equation (4.1) must be supplemented with an initial condition of the form

$$u(x,0) = f_0(x), \tag{4.2}$$

and boundary condition

$$\mathscr{B} u(x,t) = f(x,t), \quad x \in \partial\Omega, t > 0, \tag{4.3}$$

where $\mathscr{B}$ can be a Dirichlet, a Neumann or a mixed boundary operator. If we use equation (3.11) for the differentiation matrices, assume Dirichlet boundary conditions, and collocate the PDE (4.1) at the node points, we get the system of ODEs

$$U_I'(t) = \kappa W_{\Delta,I} U_I(t) + v \cdot W_{\nabla,I} U_I(t) + F(t), \tag{4.4}$$

where $W_{\cdot,I}$ contains the columns of the differentiation matrix corresponding to interior nodes and $W_{\cdot,b}$ contains the columns of the differentiation matrix corresponding to the boundary points. The matrix $W_{\nabla,I}$ is vector valued and the dot product with the velocity $v$ should be taken for each node point. The vector $U_I(t) = [u_1(t), \ldots, u_{N_I}(t)]^T$ and the vector $F(t) = \Big(\kappa W_{\Delta,b} + v \cdot W_{\nabla,b}\Big) \underline{f}(t)$, where $\underline{f}(t) = [f(x_{N_{I+1}}, t), \ldots, f(x_N, t)]^T$.

We can solve the system of ODEs in equation (4.4) in MATLAB with the ODE solver command `ode15s`, which is suitable for stiff ODEs or implement some other common time stepping method.

### *4.1 Error estimate*

In the calculation of an upper bound for the discrete error, we need the following three functions: the exact solution $u(x,t)$, the RBF approximation $s(x,t)$ from (3.10), and the auxiliary function $z(x,t)$, which interpolates the exact solution at each time

$$z(x,t) = \sum_{j \in I(x)} \sum_{k \in J(\Omega_j)} (w_j(x) \psi_k(x)) u(x_k, t) \tag{4.5}$$

The initial conditions for all three functions coincide at the collocation points. That is,

$$s(x_i,0) = z(x_i,0) = u(x_i,0), \quad 1 \le i \le N_I. \tag{4.6}$$

We define the error function $e(x,t) = u(x,t) - s(x,t)$ and the interpolation error $\varepsilon(x,t) = u(x,t) - z(x,t)$. The time evolution of the error is governed by the PDE

$$\begin{aligned} \frac{\partial e(x,t)}{\partial t} &= \frac{\partial u(x,t)}{\partial t} - \frac{\partial s(x,t)}{\partial t} \\ &= \mathscr{L} u(x,t) - \mathscr{L} s(x,t) \\ &= \mathscr{L} z(x,t) - \mathscr{L} s(x,t) + \mathscr{L} u(x,t) - \mathscr{L} z(x,t) \\ &= \mathscr{L} z(x,t) - \mathscr{L} s(x,t) + \mathscr{L} \varepsilon(x,t). \end{aligned} \tag{4.7}$$

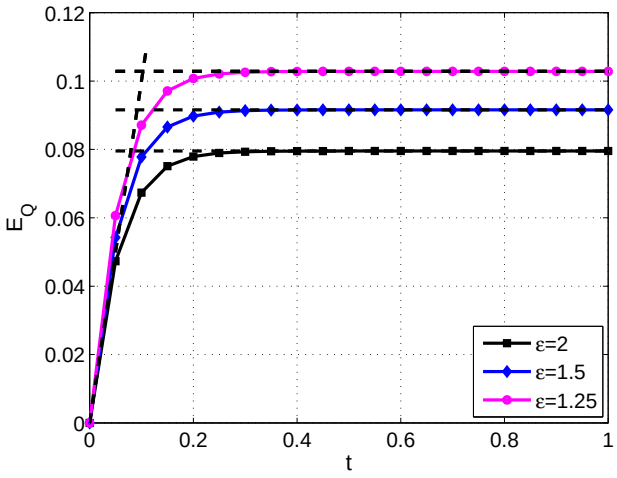


Fig. 1. The variation of $E_Q$ with time. Initially the value is close to $t$ and then approaches the asymptotic value of $\|Q^{-1}\|$. The asymptotic results are indicated by the dashed trend lines.

We will derive an estimate for the discrete error. Therefore, we define the vectors $E(t) = [e(x_1,t),\ldots,e(x_{N_I},t)]^T$ and $\varepsilon_{\mathscr{L}}(t) = [\mathscr{L}\varepsilon(x_1,t),\ldots,\mathscr{L}\varepsilon(x_{N_I},t)]^T$, and evaluate (4.7) at the interior collocation points to get

$$E'(t) = (\kappa W_{\Delta,I} + v\cdot W_{\nabla,I})E(t) + \varepsilon_{\mathscr{L}}(t), \tag{4.8}$$

where we have used the fact that $z(x_i,t) = u(x_i,t)$ for all node points, and that both $z(x,t)$ and $s(x,t)$ are PU approximations in the forms (4.5) and (3.10). Let $Q = \kappa W_{\Delta,I} + v\cdot W_{\nabla,I}$. Then, the system of ODEs (4.8) can be formally integrated to yield

$$E(t) = \int_0^t e^{Q(t-\tau)}\varepsilon_{\mathscr{L}}(\tau)d\tau. \tag{4.9}$$

A simple worst case estimate for the discrete error becomes

$$\|E(t)\|_\infty = \left\| \int_0^t e^{Q(t-\tau)}\varepsilon_{\mathscr{L}}(\tau)d\tau \right\|_\infty \le \|Q^{-1}(I-e^{Qt})\|_\infty \max_{0\le\tau\le t}\|\varepsilon_{\mathscr{L}}(\tau)\|_\infty \equiv E_Q \max_{0\le\tau\le t}\|\varepsilon_{\mathscr{L}}(\tau)\|_\infty \tag{4.10}$$

We will investigate the size of $E_Q$ further. For small enough $t$, we can Taylor expand the matrix exponential as $e^{Qt} = I + tQ + \frac{t^2}{2}Q^2 + \mathscr{O}(t^3Q^3)$, which leads to

$$Q^{-1}(I-e^{Qt}) = -\left(t + \frac{t^2}{2}Q + \mathscr{O}(t^3Q^2)\right).$$

For large enough values of $t$, we instead use the form

$$Q^{-1}(I-e^{Qt}) = V_Q\Lambda^{-1}(I-e^{\Lambda t})V_Q^{-1}, \tag{4.11}$$

where the diagonal matrix $\Lambda$ contains the eigenvalues $\lambda_j$ of $Q$ and the columns of the matrix $V_Q$ are the eigenvectors of $Q$. Numerical experiments indicate that all eigenvalues $\lambda_j$ have a negative real part. Therefore, the exponential in (4.11) approaches zero as time increases, and the limit value of $E_Q$ becomes $\|Q^{-1}\|_\infty$. Figure 1 shows how $E_Q$ changes over time for the convection diffusion example. We can see that the asymptotic value is small, that it depends on the shape parameter, and that it is reached quickly (for this time scale). In order to investigate if $\|Q^{-1}\|_\infty$ also depends on the discretization, we define the local fill distance for nodes

$$h_j = \sup_{x\in\Omega_j}\min_{k\in J(\Omega_j)}\|x-x_k\|, \tag{4.12}$$

which can be explained as measuring the radius of the largest ball empty of nodes in partition $j$, the global fill distance

$$h = \max_{1\le j\le M} h_j, \tag{4.13}$$

and the partition fill distance

$$H = \sup_{x\in\Omega} \min_{1\le j\le M} \|x - \mathbf{x}_j\|, \tag{4.14}$$

which similarly measures how densely the partition centers $\mathbf{x}_j$ cover the domain. For uniform discretizations, $h$ is proportional to the node distance and $H$ to the partition size. Figure 2 shows how the maximum $E_Q$ value depends on $h$ and $H$. The figure shows a weak dependence on $h$, especially in the more resolved cases, and an even weaker dependence on $H$.

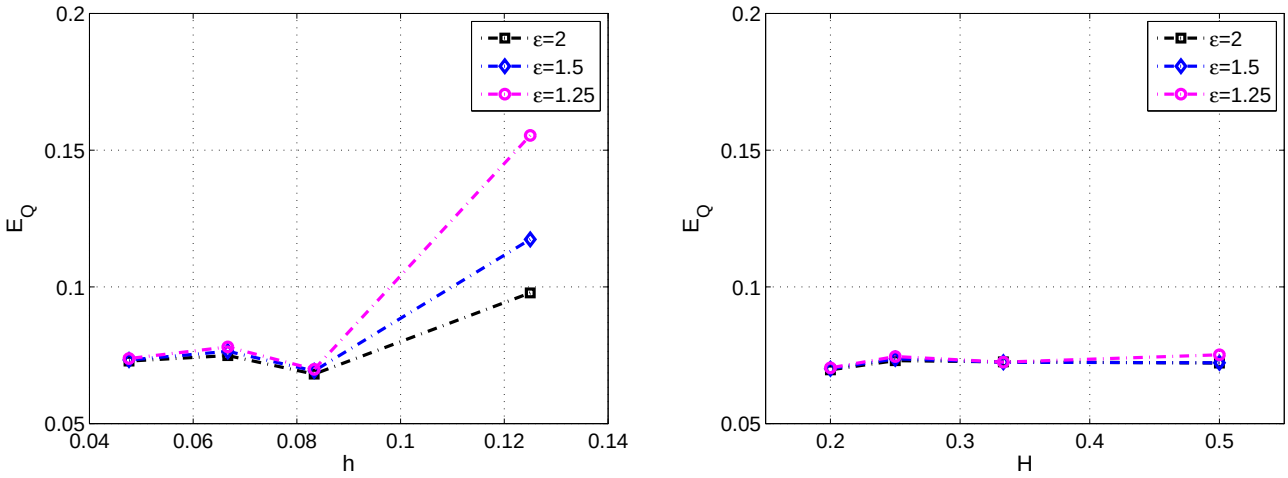


Fig. 2. The dependence of $E_Q$ on the fill distance $h$ when $H = 0.2$ (left) and the dependence on the partition fill distance $H$ when $h = 0.077$ (right) for different values of the shape parameter $\varepsilon$.

We now change our focus to the second part of the error estimate (4.10) and expand this to get

$$\|E(t)\|_\infty \le E_Q \left( \kappa \max_{0\le\tau\le t} \|\varepsilon_\Delta(\tau)\|_\infty + d\|v\|_\infty \max_{0\le\tau\le t} \|\varepsilon_\nabla(\tau)\|_\infty \right). \tag{4.15}$$

The RBF–PU interpolation error and its derivatives have been discussed extensively in [15] and rely on the sampling inequalities derived in [21]. We choose to consider two different modes of refinement in order to separate the dependence on $h$ and $H$ in the error estimates. For the first refinement mode, we require the number of nodes per partition to stay the same while we refine the partitions, meaning that $h = cH$, where $c < 1$ is a constant. Then we have the following estimate from [15] for a derivative of the interpolation error

$$\|\varepsilon_{\mathscr{L}}\|_\infty \le C \max_j C_j r_j^{m_j - \frac{d}{2} - \alpha_{\mathscr{L}}} \|u\|_{\mathscr{N}(\Omega_j)}, \tag{4.16}$$

where $r_j$ is the radius of partition $j$, $m_j$ depends on the number of nodes within the partition, $\alpha_{\mathscr{L}}$ is the order of the highest derivative in the operator $\mathscr{L}$, and $\|\cdot\|_{\mathscr{N}(\cdot)}$ denotes the native space norm (cf. [4,21]) corresponding to the chosen radial basis or kernel function. For the other refinement mode, we fix the partition and then change the number of node points locally or globally, leading to the following convergence estimate

$$\|\varepsilon_{\mathscr{L}}\|_\infty \le C \max_j e^{\gamma_j \log(h_j)/\sqrt{h_j}} \|u\|_{\mathscr{N}(\Omega_j)}. \tag{4.17}$$

When the estimates are expressed on this form, we clearly see the potential for adaptive refinement in relation to the local behavior of the solution. However, when performing numerical convergence studies, we work with (quasi) uniform discretizations, in which case $r_j$ can be replaced by $H$ and $h_j$ by $h$.

Combining the different parts of the error estimate, we get the following estimates for the two different refinement cases for uniform discretizations

$$\|E(t)\|_\infty \leq CE_Q H^{m-\frac{d}{2}-2} \max_{0\leq\tau\leq t} \max_j \|u(\tau)\|_{\mathcal{N}(\Omega_j)}, \tag{4.18}$$

$$\|E(t)\|_\infty \leq CE_Q e^{\gamma \log(h)/\sqrt{h}} \max_{0\leq\tau\leq t} \max_j \|u(\tau)\|_{\mathcal{N}(\Omega_j)}, \tag{4.19}$$

where the constants $m$ and $\gamma$ that determine the order of or rate of convergence, are taken as the minimum values over all partitions. We conclude that we expect to observe algebraic convergence in $H$, when the number of nodes per partition is fixed, and spectral convergence in $h$ when the partitions are fixed.

*Remark:* When the shape parameter $\varepsilon$ is small, the local RBF approximation is close to polynomial [14], and assuming that the node set is polynomially unisolvent, the rate constant $m$ approximately relates to the multi-variate polynomial degree $K$ supported by the number of node points within the partition as $m = K+1$. As an example, ten degrees of freedom/nodes in two dimensions corresponds to a polynomial of degree 3, leading to $m = 4$ and an overall convergence rate of $H^1$.

*Remark:* The spectral estimate involves $\sqrt{h}$ instead of $h$. This has to do with boundary effects and can be mitigated if the nodes are distributed more densely near the boundary of the approximation domain [22]. This is not practical in the PU case, since it would mean refining nodes near all partition boundaries. However, there are other effects that may in practice suppress the boundary errors, such as the decay of the weight functions towards the partition boundary.

### *4.2 Numerical results*

With appropriate initial condition and Dirichlet boundary conditions, the following function is a solution to the unsteady convection-diffusion equation (4.1) in $d = 2$ space dimensions

$$u(x,y,t) = a\exp^{bt-c(x+y)}, \tag{4.20}$$

where $a$ and $b$ can be chosen freely, and $c = \frac{v\pm\sqrt{v^2+4b\kappa}}{2\kappa} > 0$.

We discretize the domain $\Omega = [0,1]\times[0,1]$ uniformly by $N = n^2$ nodes. We also consider a corresponding non-uniform discretization with a similar number of node points. The discretizations of the square domain $\Omega$ are shown in Figure 3. Circular patches are used for partitioning of the domain. We let the overlap of the patches be 20% of the distance between the centers. An example of partitions for the square domain $\Omega$ is shown in Figure 4.

Numerical experiments are performed with the parameters $a = 1$, $b = 0.1$, $\kappa = 1$ and $v = (1,1)$. Figure 5 displays the absolute error for the two different types of node distributions shown in Figure 4 with $\varepsilon = 1.25$ at $t = 1$. The eigenvalues of the coefficient matrix $Q$ are plotted in Figure 6. It can be seen that all of the eigenvalues are in the left half plane. Therefore, the stability of the numerical solution is guaranteed.

The convergence of the solution is investigated in Figure 7 from two vantage points. In the first scenario, for a fixed number of partitions, uniformly distributed nodes with varying fill distance have been taken for studying the convergence trends of the presented method. In the second scenario, circle patches ranging from $2\times 2$ to $6\times 6$ for an almost fixed number of local nodes per partition are considered. The behavior of the convergence is studied for a fixed shape parameter $\varepsilon = 1.25$. As shown in Figure 7, increasing the number of local points for a fixed number of partitions results in spectral convergence. The rate constants $\gamma$ from (4.19) are estimated from the experimental results. For the second scenario we get algebraic convergence with respect to $H$ for a fixed number of local nodes. The slopes of the lines in the right figure can be interpreted as the approximate convergence rate, and show that we can attain high order convergence. The results are even a little better than expected from (4.18). However, with only a few measurement points we cannot draw any conclusions.

The approximated solution is plotted in Figure 8. In Table 1, the accuracy of the RBF–PUM is compared for the two types of node distributions shown in figure 4. It can be seen that the RBF–PUM is stable for long times and has almost the same accuracy for the non-uniform and uniform node distributions.

For illustrating the capability of the proposed method for irregularly shaped domains, the nonconvex domain in the left part of Figure 9 is considered. The same convection-diffusion problem (4.1) as in the previous experiment is considered with the solution (4.20). The absolute error of the approximation is plotted in Figure 9 at time $t = 1$ with $\varepsilon = 0.75$.

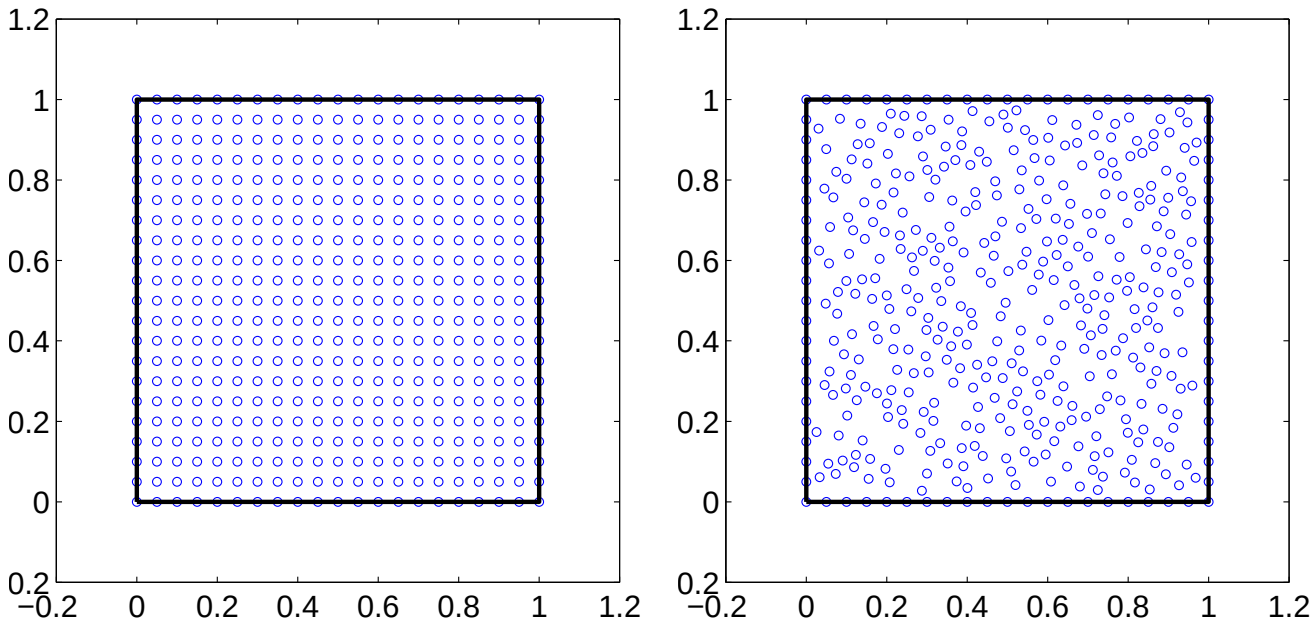


Fig. 3. Node distribution

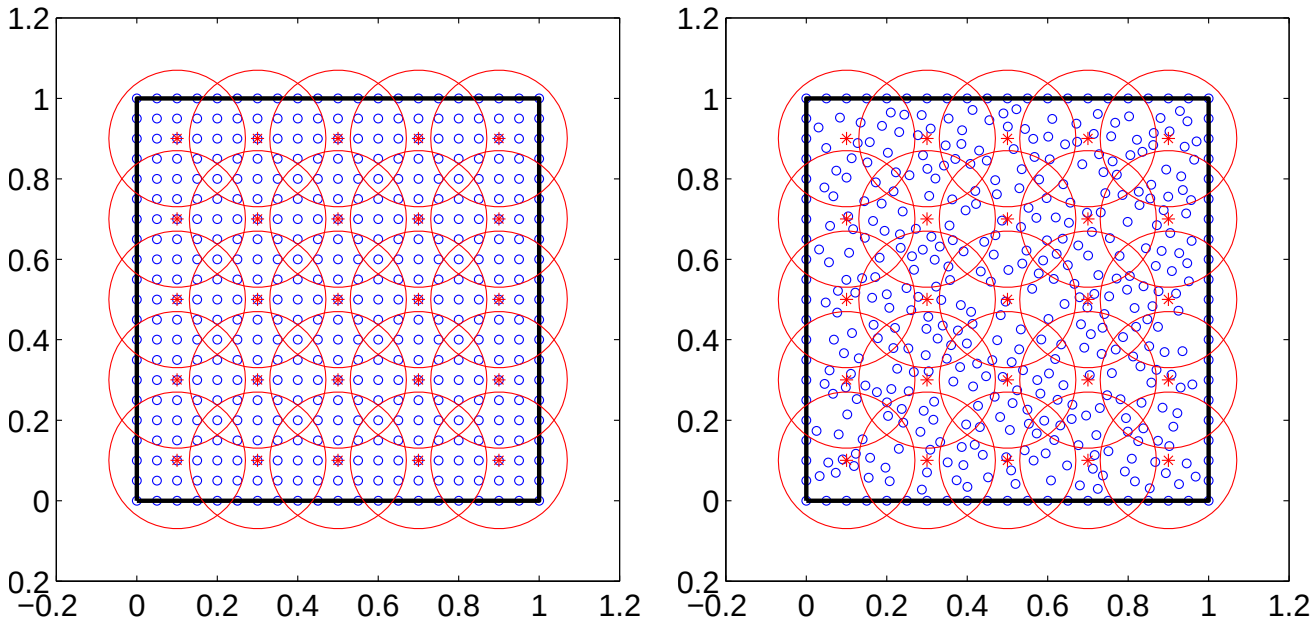


Fig. 4. Partitions of the square domain with circle patches

We have also solved the convection-diffusion problem (4.1) in three space dimensions in a solid domain bounded by the surface

$$x^2 + y^2 + z^2 - \sin(2x)^2 \sin(2y)^2 \sin(2z)^2 = 1, \tag{4.21}$$

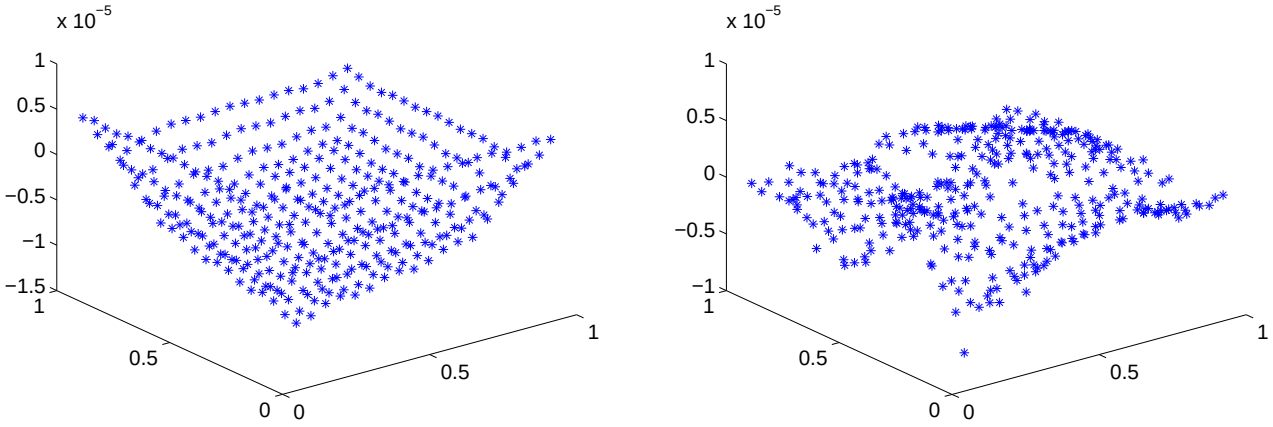


Fig. 5. Absolute error in the solution of the unsteady convection-diffusion equation with $21 \times 21$ uniform node and 445 non-uniform node points.

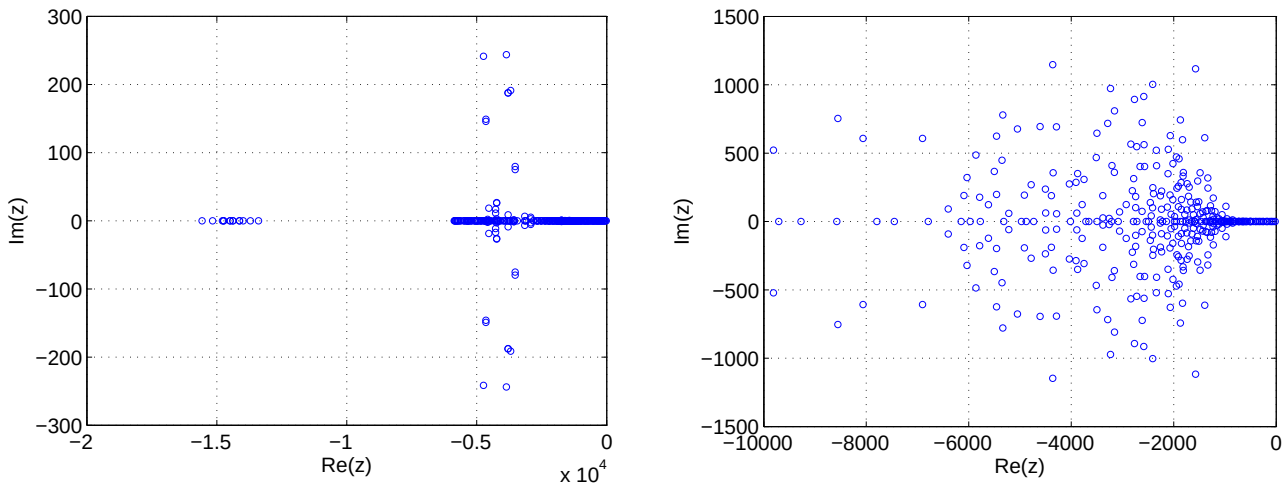


Fig. 6. Eigenvalues of the coefficient matrix $Q$ with the two types of node distributions.

Table 1
$L_\infty$ error of the RBF–PU method for $21 \times 21$ uniformly distributed nodes and 445 non-uniformly distributed nodes with $\varepsilon = 1.25$ in the square domain and 325 nodes with $\varepsilon = 0.75$ in the nonconvex domain at different times.

| $t$ | non-uniform points | uniform points | nonconvex domain |
|---|---|---|---|
| 0.1 | $7.6887e-006$ | $9.4552e-006$ | $9.9341e-006$ |
| 0.5 | $8.1349e-006$ | $1.0633e-005$ | $1.0513e-005$ |
| 1 | $8.5706e-006$ | $1.1182e-005$ | $1.9896e-005$ |
| 3 | $1.0466e-005$ | $1.3655e-005$ | $2.2720e-005$ |
| 10 | $2.1077e-005$ | $2.7511e-005$ | $3.5680e-005$ |

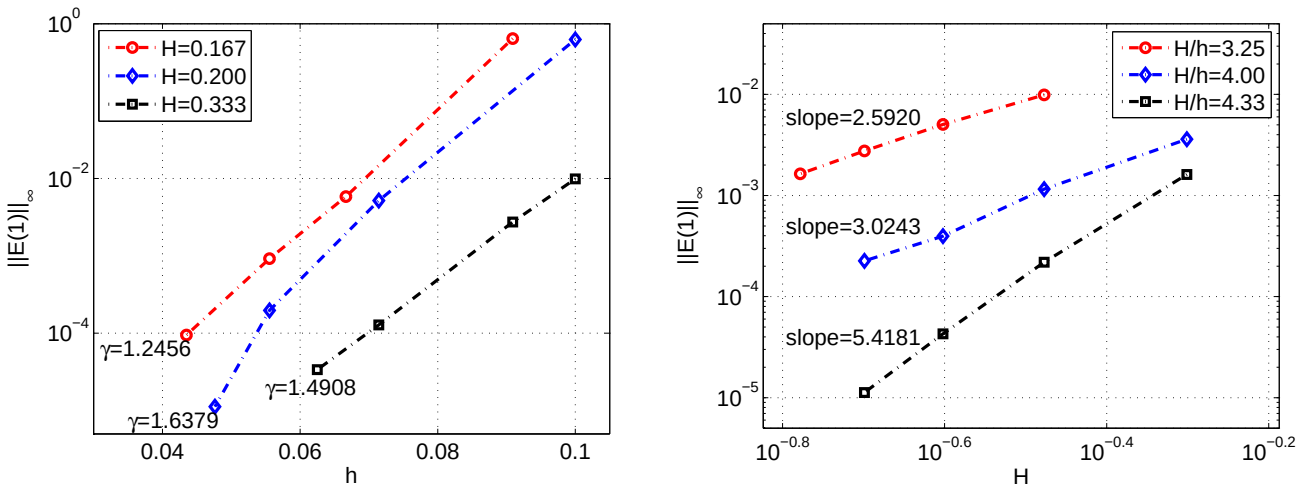


Fig. 7. The convergence as a function of $h$ for three different partition sizes $H$ (left). Convergence as a function of $H$ when the number of nodes per partition is kept (almost) fixed. The three curves correspond to approximately 21, 28, and 36 nodes per interior partition (right).

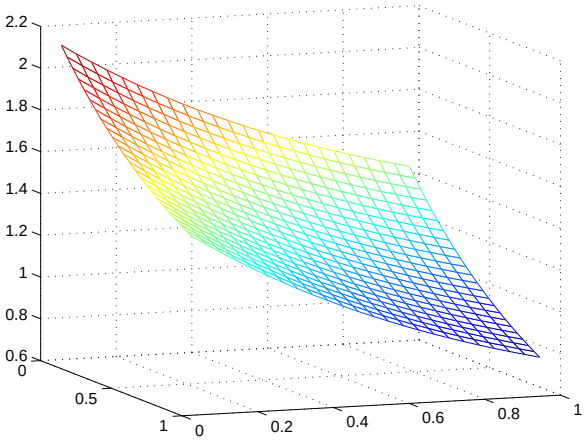

Fig. 8. Numerical solution of the unsteady convection-diffusion equation with $21 \times 21$ node points and $\varepsilon = 1.25$.

as shown in Figure 10. Boundary conditions at the surface are chosen based on the exact solution

$$u(x,y,z,t) = e^{bt-c(x+y+z)} \tag{4.22}$$

where $b = \frac{1}{10}$ and $c = \sqrt{b/6}$. With that particular choice of exact solution, the vector $\vec{v}$ in equation (4.1) can be exactly determined as $\vec{v} =< -c,-c,-c >$.

The solid domain is discretized with a total of $N = 2046$ node points and covered by $N_p = 512$ partitions. Initially, all node points are distributed uniformly. Interior nodes are then slightly perturbed in a random way in the direction towards the boundary. Non-overlapping boxes that cover the domain, which form the basis for constructing the ball covers or partitions, are shown in Figure 10.

As in the two-dimensional case, **ode15s** is used for the time-stepping. Figure 11 shows the distribution of the eigenvalues of the three-dimensional convection-diffusion operator with the number of nodes per partition $n_{\text{loc}} = 26$. Gaussian RBFs with shape parameter $\varepsilon = 0.75\overline{r}$, where $\overline{r}$ is the average radius of the ball covers, are used. All eigenvalues lie in the left half of the complex plane as is required for stability. The right subfigure in Figure 11 shows how the error evolves in time for three different values of $n_{\text{loc}}$. The growth in time is very limited and as expected the error decreases with increasing numbers of local nodes.

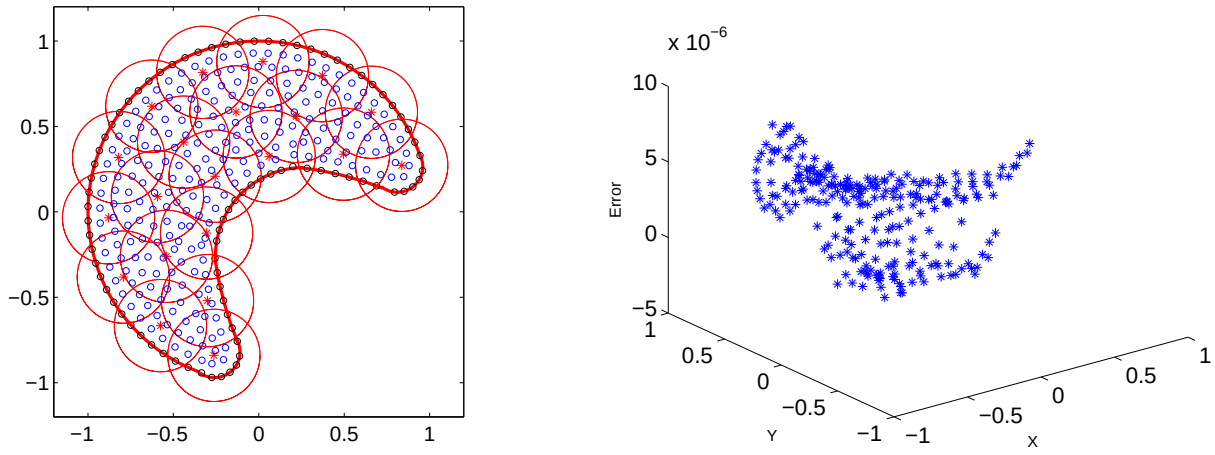


Fig. 9. Left: the partitions of the nonconvex domain with circle patches for 325 points. Right: absolute error in the solution of the unsteady convection-diffusion equation for the node distribution shown in the left figure, at time $t = 1$.

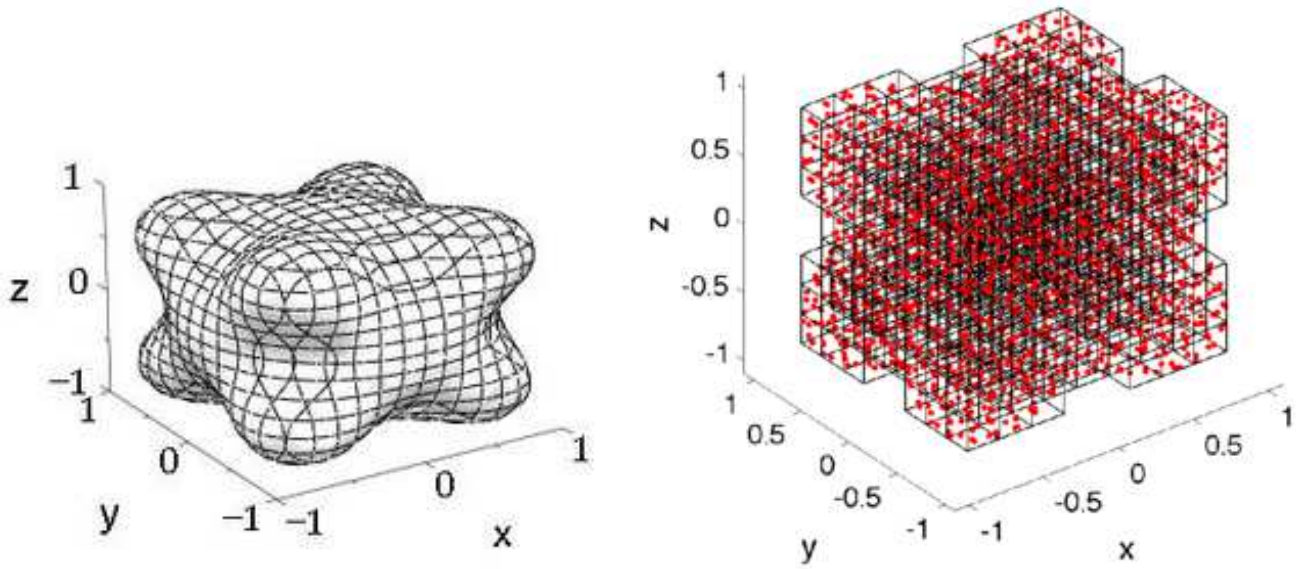


Fig. 10. The solid domain bounded by the surface equation (4.21) (left). The layout of the non-overlapping boxes which are the skeleton for the ball (partition) covers. The dots illustrate the node points (right).

## 5 Multi-asset American option pricing

The multi-dimensional version of the Black–Scholes equation takes the form

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_i S_j \frac{\partial^2 P}{\partial S_i \partial S_j} + \sum_{i=1}^{d}(r - D_i)S_i\frac{\partial P}{\partial S_i} - rP = 0, \quad 0 \le t \le T, \tag{5.1}$$

where $P$ is the value of the contract, $S_i$ is the value of the $i$th underlying asset, T is the time of expiry, $d$ is the number of underlying assets, $\rho_{ij}$ is the correlation between asset $i$ and asset $j$, $\sigma_i$ is the volatility of asset $i$, $r$ is the risk-free interest rate and $D_i$ is the (continuous) dividend yield paid by the $i$th asset. The aim is to compute $P$ in equation (5.1) throughout the time interval $[0, T]$, i.e., we will try to solve the equation (5.1) backwards in time, starting at the final condition at time $T$.

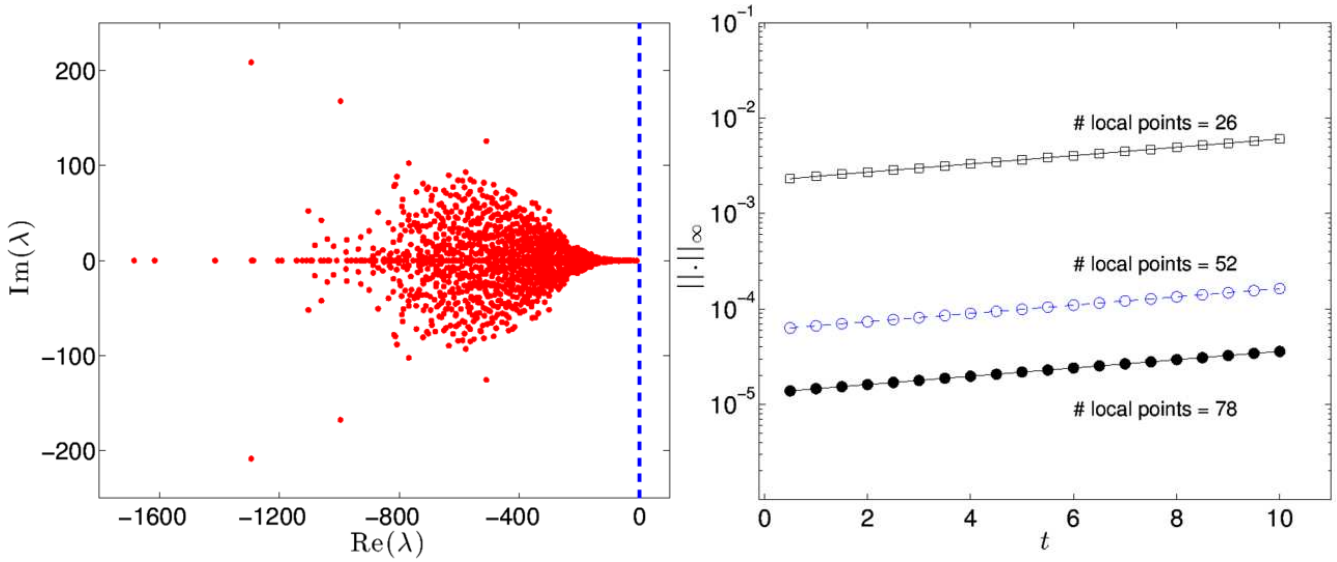


Fig. 11. Eigenvalues of the three-dimensional convection-diffusion operator discretized with Gaussian RBFs in the partition of unity setting with 26 nodes per partition (left). Error of the numerical solutions compared with the exact solution as a function of time (right).

The payoff function of the American put is given by

$$F_T(S) = \max(E - \sum_{i=1}^{d} \alpha_i S_i, 0), \tag{5.2}$$

where $E$ is the exercise price of the option and $\alpha_i$, $i = 1,\ldots,d$ are given constants. The final condition is given by

$$P(S,T) = F_T(S), \quad S \in \Omega = \mathbb{R}^d_+. \tag{5.3}$$

The boundary of the computational domain can be divided into two parts. The near-field boundary, where one or more asset prices are zero, and the far-field boundary, where one or more asset-prices tend to infinity.

For the near-field boundary, it can be noted that if one of the asset prices is zero at time $t^*$, then the asset will be worthless for any $t \geq t^*$, i.e., the solution remains at the boundary. We denote the $d$ near-field boundaries by $\Omega_i = \{S \in \Omega | S \neq 0, S_i = 0\}$, $i = 1,\ldots,d$. Then the boundary values at $\Omega_i$ can be propagated by solving a $(d-1)$-dimensional Black–Scholes problem. We denote the solutions of the reduced problems by $h_i$ and use the boundary conditions

$$P(S,t) = h_i(S,t), \quad S \in \Omega_i, \quad i = 1,\ldots,d. \tag{5.4}$$

However, already in [5] it was shown that the problem is well posed without boundary conditions at the near-field boundaries, assuming the Fichera condition on the relative strength of the drift and diffusion term holds. For a more recent discussion of the well-posedness of the problem, see also [11]. In the numerical experiments here, we will use (5.4) as in [3] even if it is not needed. For an example where near-field conditions are not used, see [20].

For put options, the contract becomes worthless as the price of any of the underlying assets tends to infinity. Therefore, we employ the following far-field boundary conditions [12]:

$$\lim_{S_i \to \infty} P(S,t) = 0, \quad S \in \Omega,\ i = 1,\ldots,d. \tag{5.5}$$

The American option allows early exercise, which means that at some values of $S$, where it is more profitable to use the option than to keep it until the expiry date, this will be done. Mathematically, this corresponds to a free boundary problem. This issue can be treated in different ways. Ito and Toivanen [10] as well as Persson and von Sydow [19] use an operator splitting

approach. The approach we will use here employs a penalty term as described in [28] and later refined in [17]. The penalty term takes the form

$$\frac{\delta C}{P+\delta-q}, \tag{5.6}$$

and ensures that the solution stays above the payoff function as the solution approaches expiry. Here $0 < \delta \ll 1$ is a small regularization parameter, $C \geq rE$ is a positive constant. The so called *barrier function* $q(S)$ is defined as

$$q(S) = E - \sum_{i=1}^{d} \alpha_i S_i, \tag{5.7}$$

see [28] for a motivation of this choice. Adding the penalty term to the Black–Scholes equation (5.1) for the American option converts it to a fixed domain problem. The penalty term is small enough so that the PDE still resembles the Black–Scholes equation closely. The error introduced by the penalty term is expected to be of the order of $\delta$. The penalty term (5.6) together with equation (5.1) lead to

$$\begin{aligned}\frac{\partial P}{\partial t} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_i S_j \frac{\partial^2 P}{\partial S_i \partial S_j} + \sum_{i=1}^{d}(r-D_i)S_i\frac{\partial P}{\partial S_i} - rP \\ + \frac{\delta C}{P+\delta-q} = 0, \quad S \in \Omega,\ 0 \leq t \leq T.\end{aligned} \tag{5.8}$$

The terminal and boundary conditions on the fixed domain are just like before

$$P(S,T) = F_T(S), \quad S \in \Omega, \tag{5.9}$$
$$P(S,t) = h_i(S,t), \quad S \in \Omega_i, \quad i = 1,\ldots,d, \tag{5.10}$$
$$\lim_{S_i \to \infty} P(S,t) = 0, \quad S \in \Omega,\ i = 1,\ldots,d. \tag{5.11}$$

Using the RBF–PUM approximation (3.11) and collocating the PDE (5.8) at the node points we get the system of ODEs

$$\begin{aligned}P_I'(t) = -\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_{iI}S_{jI}W_{ij,I}P_I(t) - \sum_{i=1}^{d}(r-D_i)S_{iI}W_{i,I}P_I(t) + rP_I(t) \\ - \frac{\delta C}{P_I(t)+\delta-q} + F(t),\end{aligned} \tag{5.12}$$

where $W_{\cdot,I}$ contains the columns of the differentiation matrix corresponding to interior points, $S_{iI}$ are diagonal matrices containing the respective coordinates of the interior node points, $P_I(t) = [P_1(t),\ldots,P_{N_I}(t)]^T$, and

$$F(t) = -\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\rho_{ij}\sigma_i\sigma_j S_{iI}S_{jI}W_{ij,b}F_b(t) - \sum_{i=1}^{d}(r-D_i)S_{iI}W_{i,b}F_b(t), \tag{5.13}$$

where $W_{\cdot,b}$ contains the columns of the differentiation matrix corresponding to boundary points and $F_b(t) = [P(x_{N_{I+1}},t),\ldots,P(x_N,t)]^T$ contains the known boundary values.

We solve the arising system of ODEs (5.12) in MATLAB with the ode solver command `ode15s`.

### *5.1 Space discretization and experimental conditions*

We note that the payoff function (5.2) possesses a discontinuity in its first derivative at the strike price. In practice, the region near the exercise price in the $(S_1,S_2)$ domain is the most interesting one where one wishes to obtain option prices. Along the $S_i$-directions we want to have a distribution of node points which is more dense in a neighborhood of the exercise price. By using an adapted node placement we aim to increase the accuracy of the approximation in the region of interest as well as to

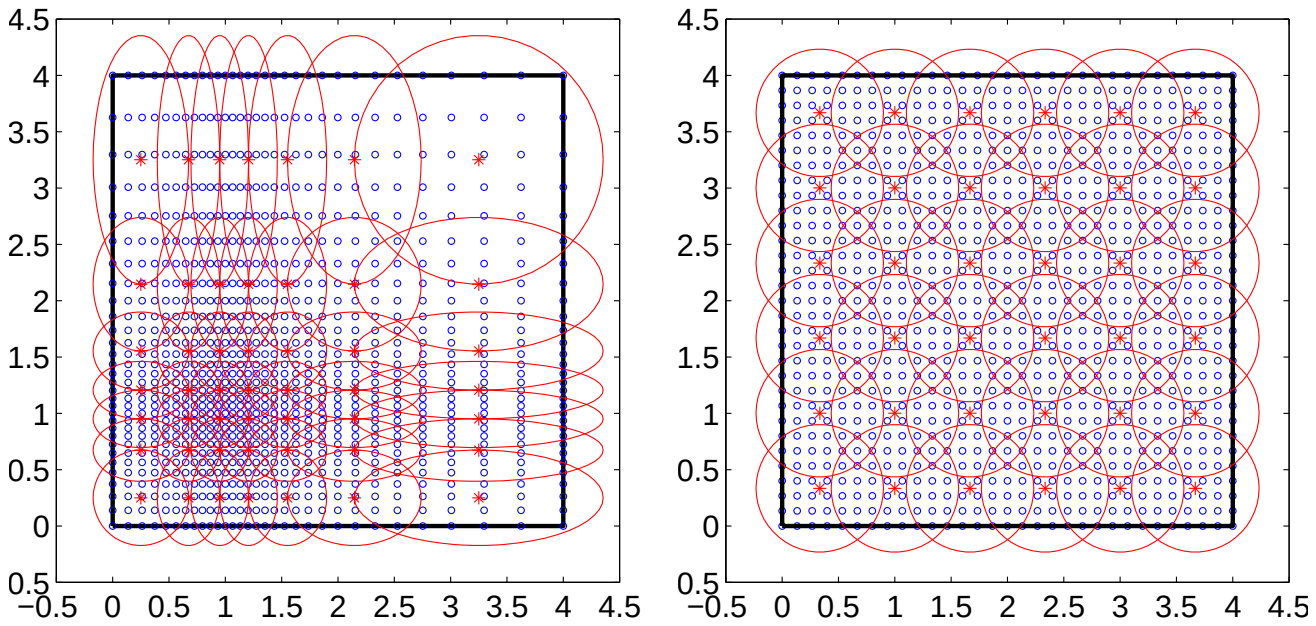


Fig. 12. Discretization of the square domain with oval and circle patches

capture the initial discontinuity in the solution better. We would like to apply the non-uniform discretization that has recently be employed, e.g., in [24,9].

In order to cluster nodes around the exercise price $E$, we define the node coordinates in each direction $i$ through

$$S_{i,j} = E + l\sinh(\xi_j), \quad 0 \le j \le m, \tag{5.14}$$

where $\xi_j \in [\xi_0, \xi_m]$ are equidistant values and $l$ is a parameter that determines the amount of clustering. By the requirement that the nodes should fall in the interval $[0, S_{i,\infty}]$ we can compute the range of $\xi$ to

$$\xi_0 = \sinh^{-1}(-E/l)$$
$$\xi_m = \sinh^{-1}((S_{i,\infty} - E)/l).$$

Note that the centers of the patches are defined with a similar pattern as for the node distribution. In our numerical experiments we have used $l = E/2$ for both nodes and partition centers.

The approximation of the solution is performed by uniform and non-uniform discretization of the square domain $\Omega = [0, S_{1,\infty}] \times [0, S_{2,\infty}]$ with oval and circle patches. Figure 12 shows the discretization pattern. For the numerical illustrations throughout this section we consider both independent colorations of the assets, i.e., $\rho_{ij} = 0$, $i \neq j$ and correlated assets with $\rho_{ij} = 0.5$, $i \neq j$. Furthermore, we use the parameter values from [3,18] given by $r = 0.1$, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\alpha_1 = 0.6$, $\alpha_2 = 0.4$, $D_1 = 0.05$, $D_2 = 0.01$ and we further take $T = 1$, $E = 1$, $S_{i,\infty} = 4E$, $L = 0$, $\delta = 0.01$ and $C = 0.2$. The Wendland function (3.3) is used for constructing the partition of unity weights, but here

$$\varphi_j(x) = \varphi\left(\sqrt{\frac{(x-x_j)^2}{r_{j,x}^2} + \frac{(y-y_j)^2}{r_{j,y}^2}}\right), \quad j = 1, \ldots, M, \tag{5.15}$$

to scale the support to fit the oval patches with radii $r_{j,x}$ in the $x$-direction and radii $r_{j,y}$ in the $y$-direction. The shape parameters for the radial basis functions used for the local approximations are scaled with respect to the node density in the patch such that

$$\varepsilon_j = \frac{\varepsilon}{\delta_j}, \tag{5.16}$$

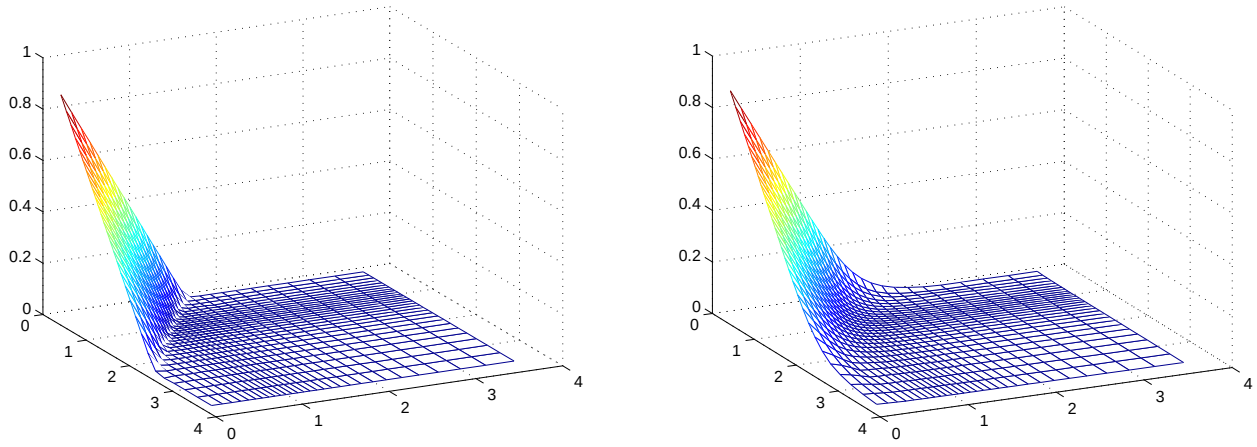

Fig. 13. The payoff function of the two-dimensional American put option (left). The approximate solution of the 2D American put option pricing at time $t = 0$ without correlation (right). The solution with correlation is visually indistinguishable.

Table 2
Approximate solution of the two-dimensional American option pricing problem for uncorrelated assets using uniform nodes with $\varepsilon = 1.5$.

| $P(S_1, S_2, t)$ | $21 \times 21$points | $31 \times 31$points | $41 \times 41$points |
|---|---|---|---|
| $P(0.9, 1, 0)$ | $1.0295e-001$ | $1.0553e-001$ | $1.0555e-001$ |
| $P(1, 0.9, 0)$ | $8.3281e-002$ | $9.1270e-002$ | $9.2725e-002$ |
| $P(1, 1.1, 0)$ | $5.2689e-002$ | $5.3921e-002$ | $5.4184e-002$ |
| $P(1.1, 1, 0)$ | $4.0733e-002$ | $4.6148e-002$ | $4.6480e-002$ |

Table 3
Approximate solution of the two-dimensional American option pricing problem for uncorrelated assets using adapted nodes.

| $P(S_1, S_2, t)$ | $16 \times 16$points | $21 \times 21$points | $26 \times 26$points | $31 \times 31$points |
|---|---|---|---|---|
| $P(0.9, 1, 0)$ | $1.0697e-001$ | $1.0608e-001$ | $1.0543e-001$ | $1.0554e-001$ |
| $P(1, 0.9, 0)$ | $9.4548e-002$ | $9.3233e-002$ | $9.2458e-002$ | $9.2615e-002$ |
| $P(1, 1.1, 0)$ | $5.5743e-002$ | $5.4418e-002$ | $5.4003e-002$ | $5.4129e-002$ |
| $P(1.1, 1, 0)$ | $4.8031e-002$ | $4.6825e-002$ | $4.6230e-002$ | $4.6380e-002$ |

where $\delta_j$ is the minimum node distance within the patch. This makes sense because the adaption of the node distribution is based on the local smoothness of the solution. Where there are higher derivatives, the shape parameter is larger and the nodes more dense, see also [6].

### *5.2 Numerical results*

Approximate solution values of the two-dimensional American option pricing problem with uncorrelated assets for different numbers of node points in the uniform and adapted cases are displayed in Tables 2 and 3. The numerical solution is computed using the parameter values defined in Section 5.1, and is evaluated at four points near the exercise price. The tables show that the approximated solution in both cases approach similar values when the number of nodes increases. From the tables, it can also be seen that the accuracy of the approximation for the same number of the points in the adapted node case is better than in the uniform case.

Table 4 shows the corresponding result for correlated assets. The relative differences between results for different numbers of nodes seem to behave similarly as in the uncorrelated case.

Table 4
Approximate solution of the two-dimensional American option pricing problem for correlated assets using adapted nodes.

| $P(S_1,S_2,t)$ | $16\times16$points | $21\times21$points | $26\times26$points | $31\times31$ points |
|---|---|---|---|---|
| $P(0.9,1,0)$ | $1.1753e-001$ | $1.1621e-001$ | $1.1595e-001$ | $1.1608e-001$ |
| $P(1,0.9,0)$ | $1.0538e-001$ | $1.0380e-001$ | $1.0346e-001$ | $1.0360e-001$ |
| $P(1,1.1,0)$ | $6.8154e-002$ | $6.6258e-002$ | $6.6192e-002$ | $6.6229e-002$ |
| $P(1.1,1,0)$ | $6.0041e-002$ | $5.8171e-002$ | $5.7971e-002$ | $5.8102e-002$ |

Figure 13, right subfigure, shows the approximate solution to the American option pricing using the nodes shown in Figure 12 and $\varepsilon = 0.15$.

## 6 Conclusions

We have implemented and tested RBF–PU methods for convection-diffusion problems such as those typically arising in valuation problems in computational finance. From the numerical experiments we have eigenvalue stability for this class of problems. A combination of theoretical and experimental analysis shows that the numerical errors are well behaved, and that we can achieve both spectral and algebraic convergence rates depending on the mode of refinement.

The RBF–PUM provides an appealing alternative to traditional collocation methods such as finite-difference (FD) and pseudospectral (PS) methods for the same problem. From the implementation perspective, differentiation matrices can be generated rather easily, and independently for each partition, without the restriction of particular stencil arrangements, or specific tensor product grids as in the FD and PS cases. This also provides flexibility with respect to the shape of the computational domain.

A main advantage of the RBF–PUM is that it allows for local adaptivity. Partitions can be locally refined and have shapes adapted to the local solution behavior as in our option pricing example. Furthermore, the node density in each partition can be locally adjusted. To develop the support for automatic adaptivity will be part of our future work, and we will also consider larger and higher-dimensional computational problems.

## References


[1] Babuška, I., Melenk, J.M.: The partition of unity method. Internat. J. Numer. Methods Engrg. **40**(4), 727–758 (1997)

[2] Duffie, D.: Dynamic Asset Pricing Theory. Princeton University Press (1996)

[3] Fasshauer, G., Khaliq, A.Q.M., Voss, D.A.: Using mesh free approximation for multi asset American options. in: C.S. Chen (Ed.), Mesh free methods, Journal of Chinese Institute of Engineers **27**, 563–571 (2004). Special issue

[4] Fasshauer, G.E.: Meshfree approximation methods with MATLAB, *Interdisciplinary Mathematical Sciences*, vol. 6. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ (2007). With 1 CD-ROM (Windows, Macintosh and UNIX)

[5] Fichera, G.: Sulle equazioni differenziali lineari ellittico-paraboliche del secondo ordine. Atti Accad. Naz. Lincei. Mem. Cl. Sci. Fis. Mat. Nat. Sez. I. VIII, Ser. 5 pp. 3–30 (1956)

[6] Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters in RBF interpolation. Comput. Math. Appl. **54**(3), 379–398 (2007)

[7] Franke, R., Nielson, G.: Smooth interpolation of large sets of scattered data. Internat. J. Numer. Methods Engrg. **15**(11), 1691–1704 (1980)

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

[9] In 't Hout, K.J., Foulon, S.: ADI finite difference schemes for option pricing in the Heston model with correlation. Int. J. Numer. Anal. Model. **7**(2), 303–320 (2010)

[10] Ito, K., Toivanen, J.: Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility. SIAM J. Sci. Comput. **31**(4), 2646–2664 (2009)

[11] Janson, S., Tysk, J.: Feynman-Kac formulas for Black-Scholes-type operators. Bull. London Math. Soc. **38**(2), 269–282 (2006)

[12] Kangro, R., Nicolaides, R.: Far field boundary conditions for Black-Scholes equations. SIAM J. Numer. Anal. **38**(4), 1357–1368 (2000). Electronic

[13] Kwok, Y.K.: Mathematical models of financial derivatives, second edn. Springer Finance. Springer, Berlin (2008)

[14] Larsson, E., Fornberg, B.: Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. Comput. Math. Appl. **49**(1), 103–130 (2005)

[15] Larsson, E., Heryudono, A.: A partition of unity radial basis function collocation method for partial differential equations (2013). Manuscript in preparation

[16] McLain, D.H.: Two dimensional interpolation from random data. Comput. J. **19**(2), 178–181 (1976)

[17] Nielsen, B.F., Skavhaug, O., Tveito, A.: Penalty and front-fixing methods for the numerical solution of American option problems. J. Comput. Finance **5**, 69–97 (2002)

[18] Nielsen, B.F., Skavhaug, O., Tveito, A.: Penalty methods for the numerical solution of American multi-asset option problems. J. Comput. Appl. Math. **222**(1), 3–16 (2008)

[19] Persson, J., Sydow, L.v.: Pricing european multi-asset options using a space-time adaptive fd-method. Tech. Rep. 2003-059, Dept. of Information Technology, Uppsala Univ., Uppsala, Sweden (2003). An optional note

[20] Pettersson, U., Larsson, E., Marcusson, G., Persson, J.: Improved radial basis function methods for multi-dimensional option pricing. J. Comput. Appl. Math. **222**(1), 82–93 (2008)

[21] Rieger, C., Zwicknagl, B.: Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning. Adv. Comput. Math. **32**(1), 103–129 (2010)

[22] Rieger, C., Zwicknagl, B.: Improved exponential convergence rates by oversampling near the boundary. Constr. Approx. (2013). In press

[23] Shepard, D.: A two dimensional interpolation function for irregularly space data. J. Comput. Appl. Math. pp. 517–524 (1968)

[24] Tavella, D., Randall, C.: Pricing Financial Instruments. Wiley, New York (2000)

[25] Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. **4**(4), 389–396 (1995)

[26] Wendland, H.: Fast evaluation of radial basis functions: methods based on partition of unity. In: Approximation theory, X (St. Louis, MO, 2001), Innov. Appl. Math., pp. 473–483. Vanderbilt Univ. Press, Nashville, TN (2002)

[27] Wu, Z., Hon, Y.C.: Convergence error estimate in solving free boundary diffusion problem by radial basis functions method. Engrg. Anal. Bound. Elem. **27**, 73–79 (2003)

[28] Zvan, R., Forsyth, P.A., Vetzal, K.R.: Penalty methods for American options with stochastic volatility. J. Comput. Appl. Math. **91**(2), 199–218 (1998)