A stable meshfree PDE solver for source-type flows in porous media

A stable meshfree PDE solver for source-type flows in porous media

R. Campagna, S. Cuomo, S. De Marchi,
E. Perracchione, G. Severino

Abstract

An elliptic partial differential equation with a singular forcing term, describing a steady state flow determined by a pulse-like extraction at a constant volumetric rate, is approximated by a radial basis function approach which takes advantage of decomposing the original domain. The discretization error of such scheme is numerically estimated and we also face up to instability issues. This produces an effective tool for real applications, as it is confirmed by comparison with classical grid-based approaches.

1 Introduction

Flows generated by isolated/distributed sources are frequently encountered into many branches of the physics, such as *electromagnetism* [1], *heat transfer* [2], and *diffusion* [3]. Moreover, in the context of the *fluid mechanics* in porous formations (aquifers and petroleum reservoirs), source-flows lend themselves as a very powerful diagnostic tool to determine the hydraulic properties of the formations [4, 5]. The idea is to model a well (or a battery of wells) like a source and assume the (pumped or injected) water so long to reach steady state conditions. The effect of such a stimulation upon the pressure distribution in the flow domain is recorded, and then the hydraulic properties are identified by matching the measured pressure-values against the theoretical ones [6].

Numerous studies have contributed to solving steady source-flows in geological formations. Modeling the formation as homogeneous (or at most as a sequence of homogeneous layers) is common to all these solutions, although it is *de facto* heterogeneous with the hydraulic conductivity varying in the space quite largely [7, 8]. These irregular changes have an impact upon flow [9, 10, 11] and transport [12, 13, 14, 15] taking place in porous formations. The common (and widely accepted) approach to tackle these erratic spatial variations is a stochastic framework that regards the hydraulic properties as random fields [16], therefore rendering stochastic the flow and transport [17, 18, 19, 20] equation.

In the present study, we consider a steady flow taking place into an unbounded domain. The flow field is generated by a given (i.e. deterministic) source function with finite support, and the aim here is the computation of the

hydraulic head u (energy per unit weight). This leads to an elliptic Partial Differential Equation (PDE) with a singular forcing term, which calls for some regularization. In order to tackle such a type of flow configuration in porous formations, where the radial symmetry is lost, due to the heterogeneous soil properties, numerical grid-based schemes, with fine grid level of discretizations near singularities, need very expensive meshes. In view of these considerations, mesh-free solvers seem to be preferable. In this realm, many methods, such as Radial Basis Function (RBF) collocation via Partition of Unity (PU) method, multiscale methods and RBF-Finite Difference (FD) local approaches, have already been developed, see e.g. [21, 22, 23, 24]. Here we deal with stability issues due to the ill-conditioning of the collocation matrices, arising in the application of the RBF-PU collocation method. To address this, we construct a hybrid method [25, 26] with the use of Variably Scaled Kernels (VSKs), introduced in [27] and further investigated in [28, 29]. Since VSKs work with any kernel, the so-constructed scheme turns out to be flexible, stable and accurate. We provide computable error estimates for the discretization via RBF-PU which are strongly consistent with numerical results. Moreover, as a confirm of the efficacy of the proposed procedure, comparisons with a grid-based scheme are also carried out.

Let us assume of having a well-type flow in a two-dimensional porous formation included in a bounded domain $\Omega \subset \mathbb{R}^2$. Under the hypothesis of *stationary* flow, stady-state confined homogeneous aquifer, i.e. with constant spatial hydraulic transmissivity $T(\boldsymbol{x}) = T$ and pumping volumetric rate $Q(\boldsymbol{x}) = Q$, and by modelling the source flow term by a Dirac delta pulse distribution δ, we solve:

$$\begin{cases} \nabla^2 u(\boldsymbol{x}) = -\frac{Q}{T}\delta(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = u_0, & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{1}$$

with $u(\boldsymbol{x})$ hydraulic head. The solution of (1) is sought subject to the Dirichlet boundary conditions on the frontier $\partial\Omega$. Under these assumptions, the Thiem solution furnishes the radial drawdown, in polar coordinates, $u(r) - u_0$ with respect to a reference level $u_0 = u(R)$, at a radial distance r from the pumping well. Let R be the radius of influence, a reference solution is given by the Thiem function:

$$u(r) - u_0 = \frac{Q}{2\pi T} \ln\left(\frac{R}{r}\right),$$

so that when $r \to R$ the hight $u(r)$ is close to the boundary condition u_0. (see Figure 1). From now on we set $u_0 = 0$ and $k := -Q/T$ for simplicity of notation. The guidelines of the paper are as follows. In Section 2 we present the PDE for source type flow in porous media and we model the singular source term. Section 3 is devoted to the presentation of the numerical scheme and error analysis. Numerical results are reported in Section 4 and the last section deals with conclusions.

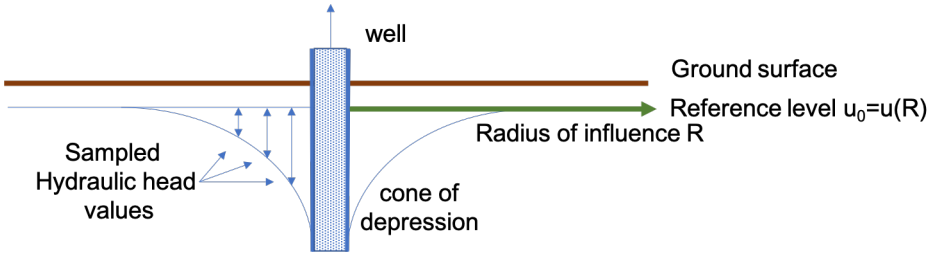

Figure 1: Problem scheme and sampling example.

2 The problem regularization

Let Ω be an open Lipschitz domain on $\mathbb{R}^M$, with polygonal boundary that contains the origin. Let $\mathcal{D}^*(\Omega)$ be the dual of the test function set $\mathcal{D}(\Omega)$, where we denote by $\mathcal{D}(\Omega)$ the space of real-valued C^∞ functions that are compactly supported on Ω and equipped with the usual topology of the test functions. Denoting by $\mathcal{H}_0^s(\Omega)$ the Sobolev space

$$\mathcal{H}_0^s(\Omega) := \{u \in L^2(\Omega) : D^{\boldsymbol{\alpha}} u \in L^2(\Omega) \quad \forall \boldsymbol{\alpha} \in \mathbb{N}^n, |\, \boldsymbol{\alpha} \,| \leq s, u\,|_{\partial\Omega} = 0\},$$

the Dirac delta $\delta \in \mathcal{D}^*(\Omega)$ can be approximated by some distributions $\tilde{\delta}_H \in \mathcal{D}^*(\Omega)$ constructed as

$$\tilde{\delta}_H(\phi) := (\delta_H, \phi)_\Omega, \qquad \forall \phi \in \mathcal{H}_0^s(\Omega), \tag{2}$$

where $\delta_H \in \mathcal{H}_0^s(\Omega)$ is a suitable *regular function* and $(\cdot,\cdot)_\Omega$ is the L^2-inner product. Moreover, after denoting by $\mathcal{H}_0^{s*}(\Omega)$ the dual of $\mathcal{H}_0^s(\Omega)$, we recall the following definition, see e.g. [30].

Definition 2.1. *Let $\delta_H \in \mathcal{H}_0^s(\Omega)$ and $\tilde{\delta}_H \in \mathcal{H}_0^{s*}(\Omega)$, defined as in (2), be a regularization of δ. We say that $\tilde{\delta}_H$ is a distribution satisfying **m-moment conditions** if and only if δ_H is compactly supported in a ball $B(\mathbf{0}, H) \subset \Omega$ and so that:*

$$(\delta_H, \mathbf{1}_\Omega)_\Omega = 1, \qquad (\delta_H, \phi^{\boldsymbol{\alpha}})_\Omega = 0, \qquad 1 \leq |\, \boldsymbol{\alpha} \,| \leq m,$$

where $\mathbf{1}_\Omega$ denotes the characteristic function of Ω.

As in [30], we focus on radially symmetric approximations, i.e.

Definition 2.2 (Radially symmetric approximation). *Let Ω be an open Lipschitz domain in $\mathbb{R}^2$ with polygonal boundary that contains the origin and $B(\mathbf{0}, H) \subset \Omega$ the ball of radius $H > 0$ centered at the origin. A **radial symmetric approximation** on $\mathcal{H}_0^s(\Omega)$ is*

$$\delta_H(\boldsymbol{x}) = \begin{cases} \dfrac{1}{H^2}\eta_m\left(\dfrac{|\boldsymbol{x}|}{H}\right), & \boldsymbol{x} \in B(\mathbf{0}, H), \\ 0, & \textit{otherwise}, \end{cases} \tag{3}$$

where $\eta_m(\boldsymbol{z}) : B(\mathbf{0}, 1) \to \mathbb{R}$ *is radially symmetric and* $m \in \mathbb{N}$ *is the number of* ***moment conditions*** *satisfied by* δ_H*.*

Finally, the problem (1) reads as follows:

Problem 2.1. *Let* $\Omega \subset \mathbb{R}^2$*,* $T(\boldsymbol{x}) = T$ *and* $Q(\boldsymbol{x}) = Q$ *be constant real functions and let* δ *be as in (3), we solve:*

$$\begin{cases} \nabla^2 u(\boldsymbol{x}) = -\frac{Q}{T}\delta_H(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = u_0, & \boldsymbol{x} \in \partial\Omega. \end{cases} \tag{4}$$

The numerical solution of Problem 2.1 is found via the method outlined in the next section.

2.1 RBF framework

We define a mesh-free scheme, based on collocation via RBFs. Such approach has been introduced by E.J. Kansa [31] and later extended to a local setting, see e.g. [32, 24]. This local scheme, which is based on collocating via the PU method, turns out to be particularly meaningful for our application, according to the arguments of the previous section. Furthermore, because of its independence of any mesh, it turns out to be easy to implement in high dimensions and flexible for different geometries; differently from classical grid-based methods such as FDs, finite elements and finite volumes.
Let Ω be a bounded domain on $\mathbb{R}^M$ and $u : \Omega \longrightarrow \mathbb{R}$. Given a set of N distinct points $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subset \Omega$, and the associated function values $\mathcal{U} = \{u(\boldsymbol{x}_1), \dots, u(\boldsymbol{x}_N)\} := \{u_1, \dots, u_N\}$, the scope in the interpolation framework is to find a function $P_{u,\mathcal{X}}$ such that:

$$P_{u,\mathcal{X}}(\boldsymbol{x}_i) = u_i, \quad i = 1, \dots, N.$$

The interpolant $P_{u,\mathcal{X}}$ is usually defined as linear combination of some *basis functions*. Given a real positive *shape parameter* ε, we here drive our attention towards strictly positive definite and symmetric kernels $K_\varepsilon : \Omega \times \Omega \longrightarrow \mathbb{R}$ so that $P_{u,\mathcal{X}} \in \text{span}\{K_\varepsilon(\cdot, \boldsymbol{x}_i), i = 1, \dots, N\}$. In other words the approximant assumes the form:

$$P_{u,\mathcal{X}}(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_k K_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_k), \quad \boldsymbol{x} \in \Omega.$$

Letting $\boldsymbol{u} = (u_1, \dots, u_N)^\intercal$ and let $A \in \mathbb{R}^{N \times N}$ be the interpolation (or kernel) matrix of entries

$$(A)_{ik} = K_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_k), \quad i, k = 1, \dots, N,$$

the coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)^\intercal$, are determined by solving the linear system $A\boldsymbol{\alpha} = \boldsymbol{u}$, which admits a unique solution provided that K is strictly positive definite and symmetric.

We further remark that, if we take radial kernels then there exists a univariate function $\phi_\varepsilon : [0, \infty) \longrightarrow \mathbb{R}$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$

$$K_\varepsilon(\boldsymbol{x}, \boldsymbol{y}) = \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{y}||_2) := \phi_\varepsilon(r).$$

We list several RBFs and their smoothness degree:

$$\begin{aligned}
&\phi_\varepsilon(r) = e^{-\varepsilon^2 r^2}, && \text{Gaussian}, && C^\infty,\\
&\phi_\varepsilon(r) = e^{-\varepsilon r}(1 + \varepsilon r), && \text{Matèrn}, && C^2,\\
&\phi_\varepsilon(r) = e^{-\varepsilon r}(15 + 15\varepsilon r + 6\varepsilon^2 r^2 + \varepsilon^3 r^3), && \text{Matèrn}, && C^6,\\
&\phi_\varepsilon(r) = (1 - \varepsilon r)_+^4 \left(4\varepsilon r + 1\right), && \text{Wendland}, && C^2,\\
&\phi_\varepsilon(r) = (1 - \varepsilon r)_+^8 \left(32\varepsilon^3 r^3 + 25\varepsilon^2 r^2 + 8\varepsilon r + 1\right), && \text{Wendland}, && C^6,
\end{aligned}$$

where $(\cdot)_+$ denotes the truncated power function. We remark that when a large number of points is involved, working with RBFs of finite regularity might be advantageous for stability purposes. Further, note that the Wendland's functions are compactly supported. We finally point out that the shape parameter, being linked to the conditioning of the kernel matrix, affects the accuracy of the solution, see e.g. [33] for further details.

For error bounds of the RBF interpolant and convergence theorems, we refer the reader to [34]. For our purposes we only remark that with the Leave One Out Cross Validation (LOOCV) scheme we are able to give error estimates by computing the residual r_i at the i-th point as ([35, 36])

$$r_i = \frac{\alpha_i}{(A)_{ii}^{-1}}, \tag{5}$$

where α_i is the i-th coefficient of the RBF interpolant $P_{u,\mathcal{X}}$ based on the full data set and $(A)_{ii}^{-1}$ is the i-th diagonal element of the inverse of the corresponding interpolation matrix. Then, as estimate of the error one can consider

$$\boldsymbol{r} = ||(r_1, \ldots, r_N)||_p,$$

where usually $p = 2, \infty$.

Dealing with applications, the conditioning of the interpolation matrix is usually high and leads to instability issues. To enhance stability, techniques allowing stable computations of the interpolant have already been developed. Among them, the most effective for the Gaussian kernel are RBF-QR methods and the Hilbert-Schmidt Singular Value Decomposition (HS-SVD), see e.g. [37, 38, 39]. Other approaches are based on finding low rank approximations of the kernel matrix, refer e.g. to [40, 41, 42] for further details. In view of these considerations, a stable way to compute the approximant is indeed necessary. To achieve both accuracy and stability, we consider the VSKs [27].

In this context, the key idea consists in substituting the scalar shape parameter with a *scale function* which plays the role of a density function (cf. [27, Def. 2.1]).

Definition 2.3. *Letting $\Sigma \subseteq (0, +\infty)$ and K_ε a positive definite radial kernel on $\Omega \times \Sigma$ depending on the shape parameter $\varepsilon > 0$. Given a scale function $\psi : \Omega \longrightarrow \Sigma$, we define a VSK K_ψ on Ω as*

$$K_\psi(\boldsymbol{x}, \boldsymbol{y}) := K_1((\boldsymbol{x}, \psi(\boldsymbol{x})), (\boldsymbol{y}, \psi(\boldsymbol{y}))),$$

for $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.

In other words, ψ defines a map

$$\Psi : \boldsymbol{x} \mapsto (\boldsymbol{x}, \psi(\boldsymbol{x})),$$

so that the VSK interpolant on the set of nodes $\Psi(\mathcal{X}) := \{(\boldsymbol{x}_k, \psi(\boldsymbol{x}_k)),\ \boldsymbol{x}_k \in \mathcal{X}\}$, with fixed shape parameter $\varepsilon = 1$, takes the form

$$P_{u,\Psi(\mathcal{X})}(\Psi(\boldsymbol{x})) = \sum_{k=1}^{N} \alpha_k K_1(\Psi(\boldsymbol{x}), \Psi(\boldsymbol{x}_k)),$$

with $\boldsymbol{x} \in \Omega,\ \boldsymbol{x}_k \in \mathcal{X}$.

The error and stability analysis of this varying scale process on Ω coincides with the analysis of a fixed scale kernel on $\Omega \times \Sigma$. However, it is worth noting that in this setting both *fill* and *separation* distances, respectively defined by

$$h_{\mathcal{X}} = \sup_{\boldsymbol{x} \in \Omega} \left(\min_{\boldsymbol{x}_k \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}_k\|_2 \right), \qquad \text{and} \qquad q_{\mathcal{X}} = \frac{1}{2} \min_{i \neq k} \|\boldsymbol{x}_i - \boldsymbol{x}_k\|_2 ,$$

are larger than in the fixed scale framework on Ω (see [27]). The former indicates how well the data fill out the domain Ω and is a measure of the interpolation error since it decreases according to it. The separation distance instead represents the radius of the largest ball that can be centred at every point in $\mathcal{X}$ such that no two balls overlap and is a measure of stability, indeed it usually decreases when the ill-conditioning grows. Therefore, the VSKs introduce a trade-off between accuracy and stability. To partially overcome this, we introduce the PU method. Taking advantage of the local scheme we can introduce a hybrid method that enables us to construct an accurate and stable numerical tool [25].

To introduce the local approach, we need to remark that the PU method partitions the bounded domain Ω into d subdomains or patches Ω_j, such that $\Omega \subseteq \cup_{j=1}^{d} \Omega_j$, with some mild overlap among them. For each of these patches, a local kernel-based approximant is defined and the global solution is constructed taking into account all the local contributions which are glued together via a family of compactly supported, non-negative and continuous functions w_j, with $\text{supp}(w_j) \subseteq \Omega_j$ and such that they form a partition of unity. Here, those weights are constructed via the Wendland's C^2 function and we take balls on $\mathbb{R}^M$ as subdomains. Their radius will be selected so that the covering properties are satisfied.

In some cases the scaling via the shape parameter provides more accurate approximations, while when the conditioning of the kernel matrices truly grows

it might be advantageous to use VSKs. Thus, given a fixed tolerance τ, we define our hybrid PU interpolant, namely HVSK-PU, as

$$I_{u,\mathcal{X}}(\boldsymbol{x}) = \sum_{j=1}^{d} w_j(\boldsymbol{x}) P_{u_j,\mathcal{X}_j}(\boldsymbol{x}), \tag{6}$$

where the local approximant $P_{u_j,\mathcal{X}_j}$ on a given Ω_j is so that

$$P_{u_j,\mathcal{X}_j}(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{k=1}^{N_j} \alpha_k^j K_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_k^j), & \boldsymbol{x} \in \Omega, \quad \sigma_m^j \geq \tau, \\ \displaystyle\sum_{k=1}^{N_j} \alpha_k^j K_\psi(\boldsymbol{x}, \boldsymbol{x}_k^j), & \boldsymbol{x} \in \Omega, \quad \sigma_m^j < \tau, \end{cases} \tag{7}$$

where $i, k = 1, \ldots, N_j$, N_j indicates the number of points on Ω_j, $\boldsymbol{x}_k^j \in \mathcal{X}_j = \mathcal{X} \cap \Omega_j$, with $k = 1, \ldots, N_j$, and σ_m^j is the smallest singular value of the local kernel matrix in the fixed scale setting whose entries are given by $K_\varepsilon(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j)$, $i, k = 1, \ldots, N_j$. In this way, we use the VSKs only when the interpolation matrix constructed via the fixed scale parameter is numerically close to be singular.

In this hybrid setting, we deal with local kernel matrices defined as:

$$(\Phi_j)_{ik} = \begin{cases} K_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_k), & \sigma_m^j \geq \tau, \\ K_\psi(\boldsymbol{x}_i, \boldsymbol{x}_k), & \sigma_m^j < \tau, \end{cases} \tag{8}$$

with $i, k = 1, \ldots, N_j$, $j = 1, \ldots, d$.

We point out that, also when using VSKs, the error of the PU interpolant can be bounded by the worst local error, indeed for $\boldsymbol{x} \in \Omega$, we have that

$$|u(\boldsymbol{x}) - I_{u,\mathcal{X}}(\boldsymbol{x})| \leq \sum_{j=1}^{d} \left|u(\boldsymbol{x}) - P_{u_j,\mathcal{X}_j}(\boldsymbol{x})\right| w_j(\boldsymbol{x}) \leq \max_{j=1,\ldots,d} \left\|u - P_{u_j,\mathcal{X}_j}\right\|_{L_\infty(\Omega_j)}. \tag{9}$$

3 The collocation scheme and accuracy results

The problem (4) is collocated via the local PU method on a global set of points $\mathcal{X} = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\}$, composed by both boundary and interior points. Thus, see e.g. [43, 24],

$$\begin{array}{rcccll} \nabla^2 I_{u,\mathcal{X}}(\boldsymbol{x}_i) & = & \sum_{j=1}^{d} \nabla^2 \left(w_j\left(\boldsymbol{x}_i\right) P_{u_j,\mathcal{X}_j}\left(\boldsymbol{x}_i\right)\right) & = & k\tilde{\delta}_H(\boldsymbol{x}_i), & \boldsymbol{x}_i \in \Omega \backslash \partial\Omega, \\ H_{u,\mathcal{X}}(\boldsymbol{x}_i) & = & \sum_{j=1}^{d} w_j\left(\boldsymbol{x}_i\right) P_{u_j,\mathcal{X}_j}\left(\boldsymbol{x}_i\right) & = & u_0, & \boldsymbol{x}_i \in \partial\Omega. \end{array}$$

Letting $\boldsymbol{\alpha}_j = (\alpha_1^j, \ldots, \alpha_{N_j}^j)^\intercal$ and let $\boldsymbol{h}_j = (P_{u_j,\mathcal{X}_j}(\boldsymbol{x}_1^j), \ldots, P_{u_j,\mathcal{X}_j}(\boldsymbol{x}_{N_j}^j))^\intercal$ be the vector of local nodal values. Since the following relation holds

$$\boldsymbol{\alpha}_j = \Phi_j^{-1} \boldsymbol{h}_j,$$

we get

$$\nabla^2 \boldsymbol{h}_j = \Phi_j^{\nabla^2} \Phi_j^{-1} \boldsymbol{h}_j,$$

where, on a given Ω_j, $\Phi_j^{\nabla^2}$ is the matrix defined as

$$(\Phi_j^{\nabla^2})_{ik} = \begin{cases} \nabla^2 K_\varepsilon(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j), & \sigma_m \geq \tau, \\ \nabla^2 K_\psi(\boldsymbol{x}_i^j, \boldsymbol{x}_k^j), & \sigma_m < \tau. \end{cases} \tag{10}$$

Thus the local elliptic discrete operator is given by

$$(L_j)_{ik} = \begin{cases} (\bar{L}_j)_{ik}, & \text{for } \boldsymbol{x}_i^j \in \Omega \backslash \partial\Omega, \\ \delta_{ik}, & \text{for } \boldsymbol{x}_i^j \in \partial\Omega, \end{cases}$$

with

$$(\bar{L}_j)_{ik} = \left(\bar{w}_j^{\nabla^2} \Phi_j + 2\bar{w}_j^{\nabla} \cdot \Phi_j^{\nabla} + \bar{w}_j \Phi_j^{\nabla^2} \right) \Phi_j^{-1} := W_j \Phi_j^{-1}, \tag{11}$$

where the local differentiation matrices are constructed by substituting the proper differential operator in (10), δ_{ik} denotes the Kronecker delta, and

$$\bar{w}_j^{\nabla^2} = \operatorname{diag}\left(\nabla^2 w_j(\boldsymbol{x}_1^j), \ldots, \nabla^2 w_j(\boldsymbol{x}_{N_j}^j) \right).$$

Similarly we define $\bar{w}_j^{\nabla}$ and $\bar{w}_j$.

Finally, let $\boldsymbol{x}_{\zeta_{kj}} \in \mathcal{X}_{N_j}$ be the node corresponding to $\boldsymbol{x}_k \in \mathcal{X}$. The global discrete PDE operator is then given by

$$(L)_{ik} = \sum_{j=1}^{d} (L_j)_{\zeta_{ij}.\zeta_{kj}}, \quad i,k = 1, \ldots, N, \tag{12}$$

and to approximate the solution, we solve

$$L\boldsymbol{z} = \boldsymbol{u}, \tag{13}$$

where $\boldsymbol{z} = (H_u(\boldsymbol{x}_1), \ldots, H_u(\boldsymbol{x}_N))^\intercal$ and $\boldsymbol{u} = (u_1, \ldots, u_N)^\intercal$, with

$$u_i = \begin{cases} k\delta_H(\boldsymbol{x}_i), & \text{for } \boldsymbol{x}_i \in \Omega \backslash \partial\Omega, \\ u_0, & \text{for } \boldsymbol{x}_i \in \partial\Omega, \end{cases} \qquad i = 1, \ldots, N.$$

As a final remark, we point out that for this local collocation method, we need to distribute the points among subdomains. Many procedures have been developed for such problem. Here we use the so-called *block-based data structure* and we refer the reader to [44, 45] for further details.

3.1 Algorithmic details

Before showing error bounds and error estimates, we summarize the steps of the proposed scheme for solving the problem (1) in the `HVSK-PU` approach. In order to compute a set of approximated values:

$$\mathcal{A}_s = \{ I_{u,\mathcal{X}}(\tilde{\boldsymbol{x}}_i), i = 1, \ldots, s \},$$

the algorithm has to define a set

$$\mathcal{C}_d = \{\bar{\boldsymbol{x}}_j, j = 1, \dots, d\} \subseteq \Omega,$$

of d of PU centres. Since d should be proportional to N, here we fix [25, 33]

$$d = \left\lfloor \frac{\sqrt[M]{N_c}}{2^{M-1}} \right\rfloor^M . \tag{14}$$

with $M = 2$ and N_c interior boundary nodes, such that $N = N_c + N_b$, with N_b boundary nodes. Moreover, in order to satisfy the covering property, we construct the partition $\Omega_1, \dots, \Omega_d$ as balls of radius

$$\rho = \left(\frac{2}{d}\right)^{1/M} . \tag{15}$$

Finally the block-based partitioning data structure is used to organize nodes and evaluation points among the subdomains.

Algorithm 1 The `HVSK-PU` algorithm

{**Input**: Q, T problem (1) parameters, $\eta_{m,p}$ regularizing function;
$\mathcal{X} = \{\boldsymbol{x}_i, i = 1, \dots, N\} \subseteq \Omega$, set of data points;
d, number of PU subdomains, $\tilde{\boldsymbol{x}}_i, i = 1, \dots, s$, evaluation points;
K kernel; ε shape parameter; ψ scale function; τ, tolerance. }
{**Initialise**:}
definition of number and type of PU centers: $\mathcal{C}_d = \{\bar{\boldsymbol{x}}_j, j = 1, \dots, d\}$, d as in (14);
definition of the PU balls radius: ρ as in (15);
organize nodes and evaluation points among the subdomains via block-based partitioning data structures.
{**Computation**:}
For each subdomain $j = 1, \dots, d$:
    compute (11) by selecting the kernel family as in (10).
    accumulate the local contributions into (12).
End for
Solve (13) and evaluate $I_{u,\mathcal{X}}$.
{**Output**: $\mathcal{A}_s = \{I_{u,\mathcal{X}}(\tilde{\boldsymbol{x}}_i), i = 1, \dots, s\}$}

3.2 Method accuracy

For our scheme, the accuracy on the computed solution depends on both the *regularization error*, introduced by the approximated mathematical model, and the *discretization error*, introduced by the numerical collocation scheme. While for the former upper bounds on the error are available, for the latter we only provide error estimates which are consistent with numerical results.

Once we regularize δ_H, we obviously introduce an approximation whose convergence order can be bounded via the following theorem [30].

Theorem 3.1 (Regularization error). *Let $\mathcal{H}_0^s(\Omega)$ be a Hilbert space and $\mathcal{H}_0^{s*}(\Omega)$ its dual. Let be $\delta \in \mathcal{H}_0^{s*}(\Omega)$ the Dirac delta, there exists a regularization*

$\tilde{\delta}_H \in \mathcal{H}_0^{s*}(\Omega)$ *that satisfies*

$$\epsilon_\delta = \mid \delta(\phi) - \tilde{\delta}_H(\phi) \mid \leq C(\phi, m) H^{m+1},$$

where $\phi \in \mathcal{D}(\Omega)$, $m \in \mathbb{N}$ *is the number of moment conditions and* $C(\phi, m)$ *is a constant that depends on* ϕ *and* m.

δ_H is usually constructed by taking $\eta_m := \eta_{m,p}$ as a polynomial of degree p, satisfying the $m-$moment conditions. We list below several examples of regularizations together with their approximation order and smoothness degree.

$$\begin{aligned}
\eta_{1,2} &= \frac{12}{\pi}(5r^2 - 8r + 3), && \mathcal{O}(H^2), \quad C^0, \\
\eta_{2,3} &= -\frac{60}{\pi}(75r^3 - 15r^2 + 10r - 2), && \mathcal{O}(H^3), \quad C^0, \\
\eta_{2,5} &= \frac{84}{\pi}(24r^5 - 70r^4 + 70r^3 - 25r^2 + 1), && \mathcal{O}(H^3), \quad C^1,
\end{aligned}$$

Upper bounds for PU collocation method, being based on Kansa's collocation approaches, are not available in literature. However, taking into account cross-validation schemes and their extensions to Kansa's methods [46], we are able to give an estimate of the *collocation* error and later numerically prove the convergence of the PU collocation method.

Proposition 3.1 (**Discretization error**). *Letting* $\Omega \subset \mathbb{R}^2$, $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1,\dots,N} \subseteq \Omega$, d *number of PU subdomains; set* $(\bar{L}_j^\intercal)_{ik}$ *and* $(\Phi_j)_{ii}^{-1}$ *defined as in (8) and (11) for each* $i, k = 1, \dots, N_j$, *and* $j = 1, \dots, d$. *For the HVSK-PU collocation scheme, an error estimate* e_{discr} *via LOOCV is given by*

$$e_{discr} = \max_{j=1,\dots,d} ||E_j||_p, \qquad (E_j)_{ik} = \frac{(\bar{L}_j^\intercal)_{ik}}{(\Phi_j)_{ii}^{-1}}, \tag{16}$$

where usually $p = 2$ *or it is the Frobenius norm, and* $i, k = 1, \dots, N_j$.

Proof. On a given Ω_j, the local differentiation matrix is given by

$$\bar{L}_j = W_j \Phi_j^{-1}, \tag{17}$$

and because of the symmetry of the matrix Φ_j, we can rewrite (17) as

$$\Phi_j \bar{L}_j^\intercal = W_j^\intercal.$$

Since this last equation implies that on each subdomain we deal with multiple systems having as common matrix Φ_j, taking into account (5), we define the following cost matrix:

$$(E_j)_{ik} = \frac{(\bar{L}_j^\intercal)_{ik}}{(\Phi_j)_{ii}^{-1}},$$

$i, k = 1, \dots, N_j$. Finally, taken into account (9) the upper bound for the global error estimate is given by (16). ■

In the next section we verify the reliability of the estimate (16) and the convergence of the described scheme, through its decreasing values while the number of the collocation points increases.

4 Numerical Results

In this section we show numerical results for the application of the described scheme to solve the Dirichlet problem (4) in order to emphasize the effects of some main parameters and heuristic on the accuracy. Tests were carried out with MATLAB R2018a software on an Intel(R) Core(TM) i5, 1.8 GHz processor.

Under the assumptions on the mathematical model, particularly under the hypotheses of homogeneous transmissivity, the *Thiem function* can be assumed as reference solution.

Set Halton-type *collocation points*, we compute the accuracy on the computed solution $\tilde{u}(\cdot)$ on Ne uniformly distributed *evaluation points* $\tilde{\boldsymbol{x}}_i$ by evaluating the relative Root Mean Square Error (rRMSE):

$$rRMSE = \frac{\sqrt{\frac{1}{Ne}\sum_{i=1}^{Ne} \mid u(\tilde{\boldsymbol{x}}_i) - \tilde{u}(\tilde{\boldsymbol{x}}_i) \mid^2}}{\sqrt{\sum_{i=1}^{Ne} \mid u(\tilde{\boldsymbol{x}}_i) \mid^2}}. \tag{18}$$

In the experiments, we consider the *Gaussian* and the *Matèrn* C^6 kernels. The first one is infinitely differentiable, while the second one is characterized by a finite regularity. This results in more stable solutions when using the Matèrn C^6, rather than the Gaussian function. Indeed, as well-known, the eigenvalues of the Gaussian decay very quickly, especially for a huge number of nodes, and thus the kernel matrix might be numerically non-invertible. Therefore, since in our problem a singular forcing term is involved, we definitely expect more stable results with kernels characterized by finite regularities.

As concern the regularizing term of problem (4), between the radially symmetric regularizations in 2 space dimensions [30], we choose $\eta_{1,2} \in C^0$, providing a regularization of order $\mathcal{O}(H^2)$. About the value of H, we have tested the scheme choosing H in the range $[2\rho, 6\rho]$; where ρ is the central patch radius. Particularly, since ρ decreases as N increases, according to (14) and (15), in order to verify the convergence of the numerical scheme for a fixed regularization error, we select H as the minimum value computed as $H = 4\rho$, for $N \in \{81, 289, 900, 1089\}$. Furthermore, we fix, unless otherwise noted,

- the number of Halton data sites $N = 289$ and the number of evaluation points $Ne = 40$;
- the model constant $\dfrac{Q}{2\pi T} = 120$;
- the PU weights as the Shepard's weights constructed via the Wendland's C^2 functions

- the *scale function* for the VSK technique as

$$\psi_j(\boldsymbol{x}, \boldsymbol{x}_i^j) = \sum_{i=1}^{N_j} \mid p_i^j(\boldsymbol{x}, \boldsymbol{x}_i^j) \mid \quad \text{on } \Omega_j,$$

with

$$p_i^j(\boldsymbol{x}, \boldsymbol{x}_i^j) = \frac{1}{\pi} \arctan(h_i^j(x_1 - x_{i1}^j)) \exp(-5(x_2 - x_{i2}^j)),$$

where $\boldsymbol{x} = (x_1, x_2), \boldsymbol{x}_i^j = (x_{i1}^j, x_{i2}^j) \in \Omega_j$, and $h_i^j \in \mathbb{R}^+, 1 = 1, \ldots, N_j$. In order to not further burden the computational cost, we fix $h_i^j = 7 \times 10^6, i = 1, \ldots, N_j$, and we refer the reader to [25] for further details.

The results highlight the main advantages of the presented scheme; in particular:

- the stability of the scheme with respect to the increase of the collocation points (*oversampling*). One of the advantages of the PU scheme is that it might oversample only regions where the solution has high variation, e.g. in our case the patch where the singularity is located [**?**]. However, the oversampling might cause instability problems due to ill-conditioning of the *local* collocation matrices. The VSK strategy can help to address this numerical issue. The hybrid technique uses the fixed shape parameter as long as the conditioning is acceptable and it switches to VSKs where the local matrix is close to be singular. Some results on the convergence of error estimates emphasize the effectiveness of the HVSK-PU algorithm.

- A comparison between HVSK-PU and a Finite Difference scheme concludes the section.

In Figure 2 the Halton points (blue circles) distribution is described; in the same figure are also reported those Halton points (red points) and the evaluation points (green dots) located in the central patch, where the accuracy on the regularizing term plays a crucial role. Figure 3 describes the contour plots for the Thiem and the computed solution, obtained by the radially symmetric polynomial regularization $\eta_{1,2}$ and the Gaussian kernel, with fixed shape parameter $\varepsilon = 1$. The red circle overlying the contour levels denotes the support H of the regularization $\delta_H = \frac{1}{H^2}\eta_{1,2}$. The corresponding error on the solution is $rRMSE = 1.99 \times 10^{-2}$. The condition number in Frobenius norm of the global matrix in the collocation scheme results $\kappa(L) = 2.41 \times 10^6$.
Being VSKs applied only on certain subdomains, their contribution varies for different bases functions. For example, set $\varepsilon = 1$, with the Matèrn kernel we have $rRMSE = 2.58 \times 10^{-2}$ and $\kappa(L) = 7.72 \times 10^5$. Those values highlight that the Matèrn kernel grants the same accuracy on the solution by correcting the ill-conditioning of local collocation matrices in much more patches than the Gaussian one (see Figure 4). Figure 5 describes the surfaces for the Thiem and the computed solution via the Matèrn kernel. As already mentioned, we observe that, by increasing the number of collocation points, i.e. on a set of $N = 1089$

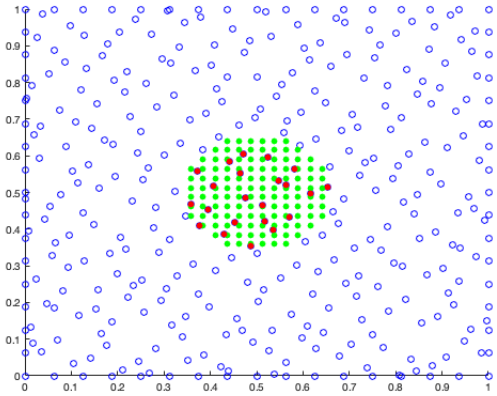

Figure 2: The Halton points (blue circles), the data in the central subdomain (red points) and the evaluation points in the central subdomain (green points).

Halton points, using the Gaussian kernel the method suffers from instability. Thus, from now on we restrict to the Matèrn C^6.

We also report in Figure 6 the sparsity pattern of the final collocation matrix. This suggests that future investigations in iterative methods for the resolution of the global system might be of interest [21]. Now we present some results confirming the robustness of the scheme as N increases. It is well known that by decreasing the fill distance the rRMSE decreases while the conditioning might increase. With $\varepsilon = 1$ and $N = 1089$ we obtain:

$$rRMSE = 7.30 \times 10^{-3} \qquad \kappa(L) = 9.01 \times 10^{6}$$

The proposed scheme tackles the increase of the collocation points in suitable subdomains. The next test is carried out with 289 data sites and thickening of 50 points in the central patch; the results prove that the accuracy is maintained:

$$rRMSE = 1.74 \times 10^{-2} \qquad \kappa(L) = 6.75 \times 10^{6}$$

In the following we explore for different ε-values ($\varepsilon = 2, 0.1$) the numerical convergence of the HVSK-PU scheme, through the *Discretization error* estimate e_{discr} in (16), which as the rRMSE should decrease as N increases. Of course, since the tolerance used to test if the local matrices are singular increases when the shape parameter decreases, as in [25], we expect that the accuracy of the HVSK-PU scheme significantly improves for small values of the shape parameters, while the solution in terms of rRMSE should be comparable with that furnished by the classica PU scheme, for $\varepsilon = 2$. Indeed, from Table 1, it is evident that rRMSEs are comparable for both methods. This is a consequence of the fact that the local matrices are invertible and thus there is no need of using any stable technique. The discretization error estimate instead is too optimistic

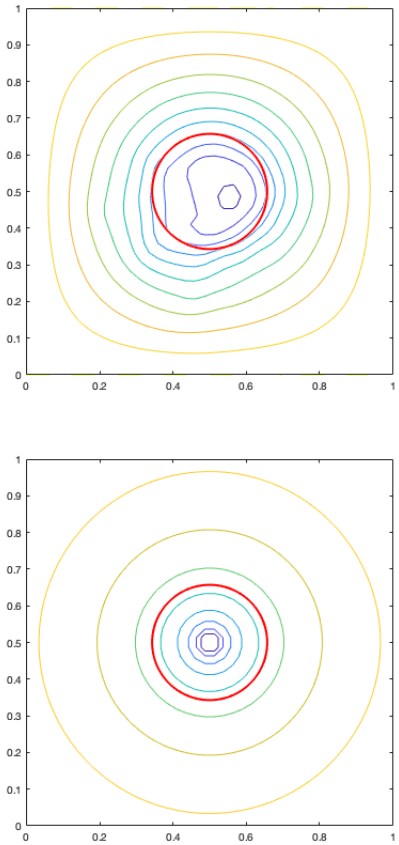

Figure 3: Contour plots of computed solution by Gaussian kernel (left) and Thiem solution (right), $N = 289$, $\varepsilon = 1$, $\eta_{1,2}$.

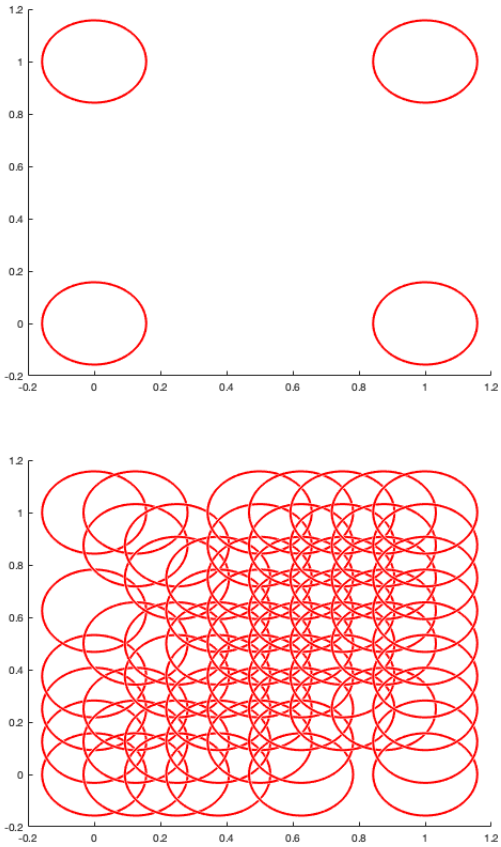

Figure 4: Distribution of the patches where the HVSK-PU algorithm dynamically applies the kernel scaling, with Gaussian (left) and Matèrn (right) kernel, respectively.

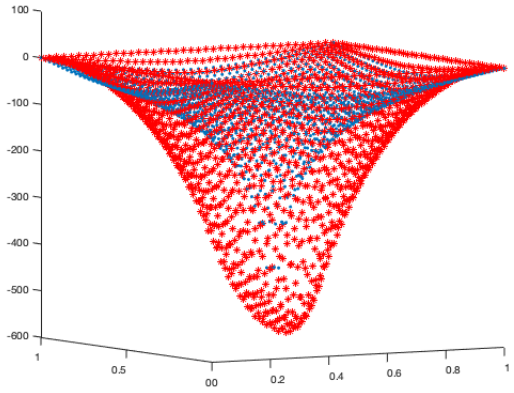

Figure 5: Computed solution by Matèrn kernel ('red *') and Thiem solution ('blue · '), $N = 289$, $\varepsilon = 1$, $\eta_{1,2}$.

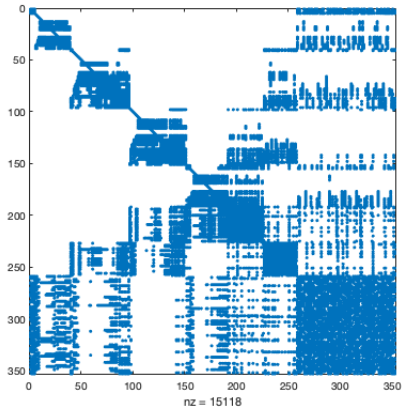

Figure 6: Sparsity of the collocation matrix on 289 data sites. nz is the number of non-zero coefficients.

for the HVSK-PU scheme, which however shows its monotonic behaviour for small values of the shape parameter, see Table 2. For this example the classical PU approach fails and this is due to the fact the ill-conditioning of the local matrices is prohibitive for stably computing their inverse. This confirms the high variability of the classical PU scheme and stresses the efficacy and convergence of the HVSK-PU approach also for small values of the shape parameters.

Finally, we compare the advantages of the HVSK-PU scheme with respect

	$\varepsilon = 2$			
nodes number	$HVSK-PU$		PU	
N	e_{discr}	$rRMSE$	e_{discr}	$rRMSE$
81	3.7533E-03	2.6213E-02	3.5232E-01	2.5191E-02
289	9.0642E-05	2.5824E-02	9.9057E-03	2.5746E-02
900	5.6812E-04	7.3459E-03	1.6099E-03	7.3392E-03
1089	1.0410E-03	7.2998E-03	1.0783E-03	7.2813E-03

Table 1: Estimation of the discretization error for classical and HVSK-PU collocation scheme for different sets of Halton data, $\varepsilon = 2$.

	$\varepsilon = 0.1$			
nodes number	$HVSK-PU$		PU	
N	e_{discr}	$rRMSE$	e_{discr}	$rRMSE$
81	3.7533E-03	2.6213e-02	8.1452E-06	1.6829E+00
289	9.0642E-05	2.5824E-02	5.3205E-02	–
900	3.7144E-05	7.3451E-03	4.5907E-02	–
1089	3.3834E-05	7.2997E-03	2.7474E-02	–

Table 2: Estimation of the discretization error for classical and HVSK-PU collocation scheme for different sets of Halton data, $\varepsilon = 0.1$.

to a FD approach for solving the elliptical problem in exam. To this end let us assume a rectangular regular domain $\Omega = [0,1] \times [0,1]$ around the singularity fixed in $P = (0.5, 0.5)$, which is covered by a regular grid of mesh-size Δ:

$$\Omega_\Delta \quad = \quad \left\{ (x_i, y_j) : \; x_i = i\Delta_x, \; i = 0, \ldots, n, \; \Delta_x = \frac{1}{n}, \; y_j = j\Delta_y, \; j = 0, \ldots, m, \; \Delta_y = \frac{1}{m} \right\} \subset \Omega, \quad m, n \in \mathbb{N}.$$

The values at the mesh points will be denoted by

$$u_{ij} \quad = \quad u(x_i, y_j), \quad i = 0, \ldots, n, \; j = 0, \ldots, m.$$

At the grid points of Ω_Δ the five-point Laplacian scheme furnishes a second order approximation for the second order derivatives:

$$\nabla^2 u(x_i, y_j) = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta^2} + \mathcal{O}(\Delta^2)$$

with $i, j = 1, \ldots, n-1$, $\Delta = \Delta_x = \Delta_y$ and $m = n$ for simplicity. The Dirac delta approximation is made in a $2\Delta \times 2\Delta$ subdomain centered at P. The FD scheme leads to the solution of a linear system of dimension n^2. Let us compare the two schemes with the same problem size. Set 81 collocation points, the Matèrn C^6 and $\varepsilon = 1$, the HVSK-PU scheme gives $rRMSE = 6.71 \times 10^{-3}$ in a time of $t = 1.15 \times 10^{-3}$ seconds, by solving a linear system with $\kappa(L) = 3.30 \times 10^4$. For

the same size the FD scheme with $\Delta = 1/8 = 0.125$ gives an approximation with relative error of order 10^{-2}, so higher than the one computed by HVSK-PU (see Table 3). In order to improve the accuracy we halve Δ so increasing the linear system dimension and consequently the computational cost of the grid-based FD scheme. The results in Table 3 confirm that the FD scheme reaches the same order of accuracy of the HVSK-PU scheme, provided many discretization points are considered and consequently with higher computational costs (of four orders greater than HVSK-PU). The $\kappa(L)$ in table refers to the conditioning of the coefficient matrix produced by the FD scheme.

n^2	Δ	rRMSE	$\kappa(L)$	t
81	0.125	8.53E-02	1.12E+03	1.68E-03
289	0.0625	4.19E-02	7.02E+03	6.65E-03
1089	0.03125	2.09E-02	4.16E+04	1.66E-01
4225	0.015625	1.04E-02	2.41E+05	7.43E+00
16641	0.0078125	6.97E-03	6.68E+05	5.80E+01

Table 3: Values obtained by FD with different mesh sizes.

5 Conclusions

In this paper we propose a numerical framework for solving an elliptic differential problem arising in hydraulic engineering. The pulse like source term, modelled by a Dirac delta function, has been firstly regularized. Then, we apply a radial basis function-based scheme which takes advantage of decomposing the original domain and allows the definition of an hybrid algorithm. The latter turns out to be very flexible and enables us to control the ill-conditioning of the collocation matrices through a kernel scaling. The discretization error for the numerical scheme is estimated and comparisons with a classical grid-based approach are also presented.

Acknowledgement

This research has been accomplished within Rete ITaliana di Approssimazione (RITA) and partially supported by the GNCS-INdAM. The fourth author has been supported by the research projects *Radial basis functions approximations: stability issues and applications*, No. BIRD167404 and *Data assimilation methods for machine learning with applications to environmental data*, founded by GeoEssential UE ERA-PLANET GA n. 689443. The authors thank Giovanna Ameno (Department of Agricultural Sciences, University of Naples - Federico II, Italy) for the valuable assistance during the literature review.

References

[1] J. D. Jackson. *Classical electrodynamics*. John Wiley & Sons, New York, 2007.

[2] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids*. Oxford Science Publications, Oxford, 2000.

[3] J. Crank. *The Mathematics of Diffusion*. Oxford Science Publications, Oxford, 2004.

[4] G. Severino, A. Santini, and A. Sommella. Steady flows driven by sources of random strength in heterogeneous aquifers with application to partially penetrating wells. *Stochastic Environmental Research and Risk Assessment*, 22(4):567–582, 2008.

[5] G. Severino. Stochastic analysis of well-type flows in randomly heterogeneous porous formations. *Water Resources Research*, 47(3), 2011.

[6] S. Gómez, G. Severino, L. Randazzo, G. Toraldo, and J.M. Otero. Identification of the hydraulic conductivity using a global optimization method. *Agricultural Water Management*, 96(3):504–510, 2009.

[7] G. Severino, A. Santini, and A. Sommella. Determining the soil hydraulic conductivity by means of a field scale internal drainage. *Journal of Hydrology*, 273(1):234 – 248, 2003.

[8] C. Fallico, S. De Bartolo, M. Veltri, and G. Severino. On the dependence of the saturated hydraulic conductivity upon the effective porosity through a power law model at different scales. *Hydrological Processes*, 30(13):2366–2372, 2016.

[9] A. Comegna, A. Coppola, V. Comegna, G. Severino, A. Sommella, and C.D. Vitale. State-space approach to evaluate spatial variability of field measured soil water status along a line transect in a volcanic-vesuvian soil. *Hydrology and Earth System Sciences*, 14(12):2455–2463, 2010.

[10] G. Severino and A. Coppola. A note on the apparent conductivity of stratified porous media in unsaturated steady flow above a water table. *Transport in Porous Media*, 91(2):733–740, 2012.

[11] G. Severino and S. De Bartolo. Stochastic analysis of steady seepage underneath a water-retaining wall through highly anisotropic porous media. *Journal of Fluid Mechanics*, 778:253–272, 2015.

[12] G. Severino, A. Comegna, A. Coppola, A. Sommella, and A. Santini. Stochastic analysis of a field-scale unsaturated transport experiment. *Advances in Water Resources*, 33(10):1188–1198, 2010.

[13] G. Severino. Macrodispersion by point-like source flows in randomly heterogeneous porous media. *Transport in Porous Media*, 89(1):121–134, 2011.

[14] G. Severino, S. De Bartolo, G. Toraldo, G. Srinivasan, and H. Viswanathan. Travel time approach to kinetically sorbing solute by diverging radial flows through heterogeneous porous formations. *Water Resources Research*, 48(12), 2012. W12527.

[15] G. Severino, A. Santini, and A. Sommella. Macrodispersion by diverging radial flows in randomly heterogeneous porous media. *Journal of Contaminant Hydrology*, 123(1):40 – 49, 2011.

[16] G. Dagan. *Flow and Transport in Porous Formation*. Springer-Verlag, New York, 1989.

[17] G. Severino, G. Dagan, and C. J. van Duijn. A note on transport of a pulse of nonlinearly reactive solute in a heterogeneous formation. *Computational Geosciences*, 4(3):275–286, 2000.

[18] G. Severino and P. Indelman. Analytical solutions for reactive transport under an infiltration–redistribution cycle. *Journal of Contaminant Hydrology*, 70(1):89–115, 2004.

[19] G. Severino, V.M. Monetti, A. Santini, and G. Toraldo. Unsaturated transport with linear kinetic sorption under unsteady vertical flow. *Transport in porous media*, 63(1):147–174, 2006.

[20] G. Severino, A. Santini, and V. M. Monetti. Modelling water flow water flow and solute transport in heterogeneous unsaturated porous media. In *Advances in Modeling Agricultural Systems*, pages 361–383. Springer, 2009.

[21] A. Heryudono, E. Larsson, A. Ramage, and L. von Sydow. Preconditioning for radial basis function partition of unity methods. *Journal of Scientific Computing*, 67(3):1089–1109, Jun 2016.

[22] O. Davydov and D. Thi Oanh. Adaptive meshless centres and rbf stencils for poisson equation. *Journal of Computational Physics*, 230(2):287 –304, 2011.

[23] G. Garmanjani, R. Cavoretto, and M. Esmaeilbeigi. A rbf partition of unity collocation method based on finite difference for initial-boundary value problems. *Computers & Mathematics with Applications*, 75(11):4066–4090, 2018.

[24] V. Shcherbakov and E. Larsson. Radial basis function partition of unity methods for pricing vanilla basket options. *Computers & Mathematics with Applications*, 71(1):185 – 200, 2016.

[25] S. De Marchi, Á. Martínez, E. Perracchione, and M. Rossini. Rbf-based partition of unity methods for elliptic pdes: Adaptivity and stability issues via variably scaled kernels. *Journal of Scientific Computing*, on line first, 2018.

[26] S. De Marchi, A. Martínez, and E. Perracchione. Fast and stable rational rbf-based partition of unity interpolation. *Journal of Computational and Applied Mathematics*, 2018.

[27] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback. Interpolation with variably scaled kernels. *IMA Journal of Numerical Analysis*, 35(1):199–219, 2015.

[28] L. Romani, M. Rossini, and D. Schenone. Edge detection methods based on rbf interpolation. *Journal of Computational and Applied Mathematics*, on line first, 2018.

[29] M. Rossini. Interpolating functions with gradient discontinuities via variable scaled kernels. *Dolomites Research Notes on Approximation*, 11:3–14, 2018.

[30] B. Hosseini, N. Nigam, and J. M. Stockie. On regularizations of the dirac delta distribution. *J. Comput. Phys.*, 305:423–447, 2016.

[31] E.J. Kansa. Application of hardy's multiquadric interpolation to hydrodynamics. *Proc. 1986 Simul. Conf.*, 4:111–117, 1986.

[32] E. Larsson, V. Shcherbakov, and A. Heryudono. A least squares radial basis function partition of unity method for solving pdes. *SIAM Journal on Scientific Computing*, 39:A2538–A2563, 02 2017.

[33] G. E. Fasshauer and J. G. Zhang. On choosing "optimal" shape parameters for rbf approximation. *Numerical Algorithms*, 45(1):345–368, Aug 2007.

[34] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.

[35] G.E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. Interdisciplinary mathematical sciences. World Scientific, 2007.

[36] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2):193–210, Nov 1999.

[37] R. Cavoretto, G. E. Fasshauer, and M. McCourt. An introduction to the hilbert-schmidt svd using iterated brownian bridge kernels. *Numerical Algorithms*, 68(2):393–422, Feb 2015.

[38] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic pdes. *Computers & Mathematics with Applications*, 46(5):891 – 902, 2003.

[39] E. Larsson, E. Lehto, A. Heryudono, and B. Fornberg. Stable computation of differentiation matrices and scattered node stencils based on gaussian radial basis functions. *SIAM Journal on Scientific Computing*, 35:A2096–A2119, 01 2013.

[40] S. De Marchi and G. Santin. Fast computation of orthonormal basis for rbf spaces through krylov space methods. *BIT Numerical Mathematics*, 55(4):949–966, Dec 2015.

[41] M. Pazouki and R. Schaback. Bases for kernel-based spaces. *Journal of Computational and Applied Mathematics*, 236(4):575 – 588, 2011. International Workshop on Multivariate Approximation and Interpolation with Applications (MAIA 2010).

[42] S.A. Sarra. The matlab radial basis function toolbox. *Journal of Open Research Software*, 5(1):8, 2017.

[43] A. Safdari-Vaighani, A. Heryudono, and E. Larsson. A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications. *Journal of Scientific Computing*, 64(2):341–367, Aug 2015.

[44] R. Cavoretto, A. De Rossi, F. Dell'Accio, and F. Di Tommaso. Fast computation of triangular shepard interpolants. *Journal of Computational and Applied Mathematics*, 2018.

[45] R. Cavoretto, A. De Rossi, and E. Perracchione. Efficient computation of partition of unity interpolants through a block-based searching technique. *Computers & Mathematics with Applications*, 71(12):2568 – 2584, 2016.

[46] G. E. Fasshauer and J. G. Zhang. On choosing "optimal" shape parameters for rbf approximation. *Numerical Algorithms*, 45(1):345–368, Aug 2007.