An Adaptive–Hybrid Meshfree Approximation Method

Leevan Ling

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

SUMMARY

It is now commonly agreed that the global radial basis functions method is an attractive approach for approximating smooth functions. This superiority does not come free; one must find ways to circumvent the associated problem of ill-conditioning and the high computational cost for solving dense matrix systems. We previously proposed different variants of adaptive methods for selecting proper trial subspaces so that the instability caused by inappropriate shape parameters were minimized. In contrast, the compactly supported radial basis functions are more relaxing on the smoothness requirements. By settling with algebraic order of convergence only, compactly supported radial basis functions method, provided the support radius are properly chosen, can approximate functions with less smoothness. The reality is that end-users must know the functions to be approximated *a priori* in order to decide which method to be used; this is not practical if one is solving a time evolving partial differential equation. The solution could be smooth at the beginning but the formation of shocks may come later in time. In this paper, we propose a hybrid algorithm making use of both global and compactly supported radial basis functions with other developed techniques for meshfree approximation with minimal fine tuning. The first contribution here is an adaptive node refinement scheme. Secondly, we apply the global radial basis functions (with adaptive subspace selection) on the adaptively generated data sites and lastly, the compactly supported radial basis functions (with adaptive support selection) that can be used as a blackbox algorithm for robust approximation to a wider class of functions and for solving partial differential equations.

Keywords: *radial basis function; adaptive greedy algorithm; data refinement; convergence proof; Burgers' equation.*

1. Introduction

Many problems in science and engineering require the determination of multivariate functions which have specified properties. These properties may consist of finitely many prescribed function values at certain points or of infinitely many conditions arising; e.g. from function approximation to solving partial differential equations (PDE). To avoid meshing and re–meshing, meshfree methods have been proposed; one of the simplest and easiest methods

[*]Correspondence to: Leevan Ling < lling@hkbu.edu.hk >

Contract/grant sponsor: The CERG Grant of the Hong Kong Research Grant Council and the FRG Grant of the Hong Kong Baptist University.

Received April 6, 2011
Revised

is to work with functions of the form

$$u(\overrightarrow{x}) = \sum_{j=1}^{N} \alpha_j \Phi(\overrightarrow{x}, \overrightarrow{x}_j), \tag{1}$$

with a *kernel* function $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. The set of points $X_N = \{\overrightarrow{x}_j\} \subset \mathbb{R}^d$ are called *centers* and have no specific structure in order to guarantee a meshfree definition. A further simplification is provided by scalar radial basis functions (RBF) ϕ (scaled by a shape parameter c) which allow kernels of the form $\Phi(\overrightarrow{x}, \overrightarrow{y}) = \phi(\frac{1}{c}\|\overrightarrow{x} - \overrightarrow{y}\|_2)$ leading to Euclidean invariance of reconstruction problems.

In the title of this paper, we use the phrase RBF *approximation* instead of *interpolation* as we will focus on overdetermined systems for approximation in the standard least-squares sense. Before carrying on, the following brief discussion ensures that all RBF interpolation theories can be applied to our setting. We will follow the ideas in [1] in which similar results are proven for the modified Helmholtz PDE.

Definition 1. *For each set of data sites $X_N := \{(x_i, y_i)\}_{i=1}^N \subset \overline{\Omega}$. Define a partition $P_N = \{\Omega_i\}_{i=1}^N$ of Ω by X_N such that $(x_i, y_i) \in \Omega_j$ if and only if $i = j$. The sequence $\{X_N\}_{N=1}^\infty$ is said to be* getting dense quasi-uniformly *if there exists two positive constants K_1 and K_2 independent of N such that*

$$K_1 Area(\Omega) \le N\, Area(\Omega_i) \le K_2 Area(\Omega), \qquad \text{for all } i = 1, \dots, N. \tag{2}$$

The purpose of the partition in the above definition is solely theoretical; the Voronoi diagram can be one of such a partition. Using different partitions will affect the constants K_1, K_2, and other constants in the convergence theory that depend on them.

Lemma 2. *Suppose $\Omega \subset \mathbb{R}^2$ is regular and bounded. Let $f \in C(\Omega)$ be square Riemann integrable over $\overline{\Omega}$ and the sequence $\{X_N\}_{N=1}^\infty$ of sets of data locations is* getting dense quasi-uniformly. *Define the $\ell^2(X_N)$–norm as*

$$\|f\|_{\ell^2(X_N)} := \sum_{(x_i,y_i)\in X_N} \big(f(x_i, y_i)\big)^2.$$

Then for sufficiently large N, there exist two positive constants K_1 and K_2 independent of N such that

$$K_1 \|f\|_{L^2(\Omega)} \le N^{-1/2} \|f\|_{\ell^2(X_N)} \le K_2 \|f\|_{L^2(\Omega)}.$$

Proof. The theorem obviously holds for $\|f\|_{L^2(\Omega)} = 0$. Consider $\|f\|_{L^2(\Omega)} \neq 0$. Under the assumptions on f and X_N, we have

$$\lim_{N\to\infty} \sum_{i=1}^{N} \big(f(x_i, y_i)\big)^2 Area(\Omega_i) = \|f\|^2_{L^2(\Omega)}. \tag{3}$$

For any given $\varepsilon > 0$, if N is sufficiently large, then

$$\int_\Omega f^2\, dx - \varepsilon < \sum_{i=1}^{N} \big(f(x_i, y_i)\big)^2 Area(\Omega_i) < \int_\Omega f^2\, dx + \varepsilon.$$

The Lemma is proven simply by picking $\varepsilon = \frac{1}{2}\|f\|^2_{L^2(\Omega)}$. □

We focus our discussion on some symmetric (conditionally) positive definite radial basis kernels $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. If we take the reproducing kernel Hilbert space associated with Φ (native space) as the trial space

$$\mathcal{U} = \mathcal{N}_\Phi := \overline{\operatorname{span}\{\Phi(\cdot, y) : y \in \Omega\}}^{\|\cdot\|_\Phi}, \tag{4}$$

where $\|\cdot\|_{\mathcal{U}} = \|\cdot\|_\Phi$ is induced from the inner product

$$\Big\langle \sum_i c_i \Phi(\cdot, \overrightarrow{x}_i),\ \sum_j d_j \Phi(\cdot, \overrightarrow{y}_j)\Big\rangle = \sum_{i,j} c_i d_j \Phi(\overrightarrow{x}_i,\ \overrightarrow{y}_j),$$

then the standard h^β-type error bound given in [2, 3], where h is the fill distance of the data and β is the smoothness of Φ, can be used. For this work, we only require that the discretized trial spaces $\mathcal{U}_n$ have certain approximation power.

Definition 3. *The discretized trial space $\mathcal{U}_n$ is said to have* ϵ–approximation power *(for small numbers $\epsilon > 0$) if for all $f \in \mathcal{U}$ there are some approximations $s_{f,\epsilon} \in \mathcal{U}_n$ with*

$$\|s_{f,\epsilon} - f\|_{L^2(\Omega)} \le \epsilon \|f\|_{\mathcal{U}}, \quad \textit{for all}\ \ f \in \mathcal{U}. \tag{5}$$

The exact expression for ϵ depends on our assumptions on the function of interest f and the chosen kernel Φ. In particular, an error bound [4] for compactly supported RBF (CSRBF) is in the exact same form as (5). If one considers the multiquadrics (MQ) or Gaussian (GA) RBF, then the exponential error bounds $\mathcal{O}(\lambda^{c/h})$-convergence $(0 < \lambda < 1)$ given in [5, 6] with respect to $L^\infty(\Omega)$ can be loosely employed as (provided f lies in the associated native spaces)

$$\|s_{f,\epsilon} - f\|_{L^2(\Omega)} \le K_\Omega \|s_{f,\epsilon} - f\|_{L^\infty(\Omega)} \le \epsilon K_\Omega \|f\|_{\mathcal{U}} \ \text{ for all }\ f \in \mathcal{U}.$$

Otherwise, the $L^p(\Omega)$ bound for MQ, radial powers and thin plate splines proven by [7] can be used. For functions in certain Sobolev spaces but not the native space of the kernel, Sobolev bounds for functions outside the native space with scattered zeros can be found in [8]. The RBF convergence theory itself is a big topic to be included in the work; see for instance in [9, 10, 11], and [12] for more results on RBF interpolation. Readers are also referred to the research monographs on RBF authored by Buhmann [13], Fasshauer [14], and Wendland [15] for recent reviews and details. The message is that the interpolant $s_{f,\epsilon}$ allows us to bring all RBF interpolation theories to RBF approximation.

Theorem 4. *Define the numerical approximation f_n on the given sets of collocation points $X_N \subset \overline{\Omega}$ and radial basis centers $X_n \subset X_N$, as*

$$f_n := \arg\min_{s \in \mathcal{U}_n} \|s - f\|_{\ell^2(X_N)}.$$

Suppose that $\{X_N\}_{N=1}^\infty$ is getting dense quasi-uniformly and $\{\mathcal{U}_n\}_{N=1}^\infty$ has decreasing $\epsilon(n)$–approximation powers as $n \to \infty$. Then, for sufficiently large N, the error estimate is given by

$$\|f_n - f\|_{L^2(\Omega)} \le K\,\epsilon\,\|f\|_{\mathcal{U}},$$

where K is a positive generic constant independent of N.

Proof. For simplicity, let K be a positive generic constant independent of n and N. By the minimization property of f_n, we have

$$\begin{aligned} \|f_n - f\|_{L^2(\Omega)} &\leq KN^{-1/2}\|f_n - f\|_{\ell^2(X_N)} \\ &\leq KN^{-1/2}\|s_{f,\epsilon} - f\|_{\ell^2(X_N)} \\ &\leq K\,\|s_{f,\epsilon} - f\|_{L^2(\Omega)} \leq K\,\epsilon\,\|f\|_{\mathcal{U}}. \end{aligned}$$

In other words, RBF approximation in the least-squares sense is as good as RBF interpolation (off by a constant factor) provided the set of N collocation points is dense enough. □

The above convergence theories provide us certain degrees of confidence. Similar least-squares convergence results on the unsymmetric RBF collocation method for PDE is numerically verified in [16]. Even though the contour-Padé algorithm in [17] allows stable computations of multiquadrics interpolants for all values of the shape parameter for moderate-scale problems, the practical issues found in RBF methods remain. For example, the problem of ill-conditioning is one of the biggest concerns to researchers; that is, the number of RBF centers n cannot be too large. Different techniques were developed to circumvent the problem of ill-conditioning; as examples, preconditioning [18], domain decomposition [19], matrix decomposition [20], and the more recently proposed localized RBF approaches [21, 22]. On the other hand, optimal RBF centers and data sites distribution are long open questions without solid solutions yet. Since the strong–form PDE collocation method can be theoretically viewed as a general interpolation method, we focus on the standard approximation problem in this paper. This preliminary work, besides its immediate application to solving PDE (i.e., the dual reciprocity method [23, 24] and the method of particular solution [25, 26]), provides us the fundamentals in solving time-dependent PDEs with shock formation, moving front, or complications of other sorts. But, first of all, the meshfree techniques must be made flexible and robust.

We propose an adaptive-hybrid algorithm trying to overcome as many difficulties as possible. Using the freedom provided by meshfree methods not only allows fast RBF evaluation as seen in the paper by Johnson [27], it also allows us to freely place data sites where needed most. In Section 2, an adaptive node refinement scheme is proposed to increase the density of data sites where needed most regions with rapid variation. A boundary principle function, which allows user-control over the density of near-boundary data sites, gives clear advantages over the previously proposed wavelet-based approach.

Next, in Section 3.1, the adaptive greedy algorithm for global RBF (GRBF) trial subspace selection is reviewed. In this paper, we use the the multiquadrics (MQ) $\phi(r) = \sqrt{1+r^2}$ only. To apply GRBF on the refined data sites, a level-dependent weight for each collocation condition is needed. The greedy algorithm adaptively selects the trial space to provide a stable solutioning process. This is carried out by searching a subset of GRBF centers (e.g. X_n) out of a larger set (e.g. X_N). For a rapidly changing solution, e.g. interior or boundary layers problem, using GRBF alone is certainly not sufficient. Moreover, the efficiency of the algorithm drops when the set of GRBF centers is getting large. CSRBF meshless algorithms with full mathematical support and easy implementation were, without a doubt, a breakthrough in the field in the late 90's. In Section 3.2, we call upon the CSRBF for further residual reduction. Strategies for picking support radius are discussed. In our numerical simulations shown in Section 4, various types of functions are being approximated by the proposed algorithm. The proposed method is shown to be robust with two sets of suggested user-defined parameters.

2. Adaptive node refinement (ANR)

The objective of adaptive node refinement (ANR) is to find the set of nodes (or data sites, collocation points, interpolation points) that capture important spatial information of a given function. The basic idea is to determine if refinement in a small subdomain (or cell) is needed. Repeat this step until no further refinement is needed. Collecting the corner points of all subcells (subject to duplicates removal) yields a set of refinement nodes for imposing collocation conditions. Similar approaches were commonly used with finite element methods and can also be found in [28] for RBFs.

A user-supplied criterion, or the so-called *principle function* $P(\cdot)$, is required to tag individual grid cells C for refinement. If the principle value of a cell $P(C)$ is greater than some tolerances τ, then the cell C is *splittable* and will be refined by splitting into four equally sized quadrants (or subcells, children, leave). For each square cell C with side length h in Ω, we employ the generic labeling of the cell's corners and center in Figure 1 *(Left)*. This section contains two main parts: the first part is devoted to the splitting algorithm for interior nodes and the second addresses that for cross-boundary nodes. The goal in this section is to design an appropriate principle that works for both interior and cross-boundary cases.

Let f be the function we want to approximate over the domain $\Omega \in \mathbb{R}^2$. Define the manifold $S_{f,\Omega} = \{(x, f(x)|\ x \in \Omega \subset \mathbb{R}^2\} \subset \mathbb{R}^3$. A good distribution of nodes should be more or less uniform on $S_{f,\Omega}$. Hence, the principle function should contain information on the surface area. Let $f_{i,j} := f(x_i, y_j)$ for $i, j \in \{0, \frac{1}{2}, 1\}$. The surface area $\int_C |f|\ d\sigma$ can be approximated using the function values at the four corners of the cell. Next, to enhance the flexibility of the algorithm, the semi-norm $\int_C |\nabla f|\ dx$ is also taken into the principle function using a five-point approximation scheme. For some user-defined *interior principle parameters* $\alpha \geq 0$, we now define the principle function $P_\alpha^{(int)}(C)$ as

$$\begin{aligned} P_\alpha^{(int)}(C) \quad := \quad & \frac{h}{2}\Big(\sqrt{h^2 + (f_{0,0} - f_{0,1})^2 + (f_{1,1} - f_{0,1})^2} \\ & \qquad + \sqrt{h^2 + (f_{0,0} - f_{1,0})^2 + (f_{1,1} - f_{1,0})^2}\ \Big) \\ & \qquad\quad + 2\alpha \left| f_{0,0} + f_{0,1} + f_{1,0} + f_{1,1} - 4 f_{\frac{1}{2},\frac{1}{2}} \right| \\ \approx & \int_C |f|\ d\sigma + \alpha \int_C |\nabla f|\ dx. \end{aligned} \tag{6}$$

Figure 1 *(Right)* shows the schematic illustration of the ANR scheme for the function $f(x, y) = e^{x+y}$. Note that the cells' areas decrease as the gradient of f increases towards the corner $(x, y) = (1, 1)$. The cell-structure we see here is also used in the multiscale kernels proposed by Opfer [29] and later be used in [30] for numerical differentiation.

When Ω is not rectangular, the use of square cells almost guarantees the existence of outside Ω cells (or their corners). If the cell is completely out of Ω, no refinement is needed. We now turn our focus to a cross-boundary cell $C \bigcap \overline{\Omega} \neq \emptyset$ in which $P_\alpha^{(int)}(C)$ is not well-defined (if f is not defined outside Ω). Similar to the situation found in level set method, we assume that any given points $(x, y) \in \mathbb{R}^2$ can be classified as *in*-or-*out* of Ω by a signed function:

$$\mathfrak{s}^\Omega(x, y) = \begin{cases} 1 & \text{if} \quad (x, y) \in \Omega, \\ 0 & \text{if} \quad (x, y) \in \partial\Omega, \\ -1 & \text{if} \quad (x, y) \in \mathbb{R}^2 \setminus \overline{\Omega}. \end{cases}$$

We denote its values at a given cell's corners and center, as in Figure 1, by $\mathfrak{s}^{\Omega}_{i,j} := \mathfrak{s}^{\Omega}(x_i, y_j)$ for $i, j \in \{0, \frac{1}{2}, 1\}$. For some user-defined *boundary principle parameters* $\beta \geq 0$, we now define the principle $P^{(bdy)}_{\beta}(C)$ as

$$P^{(bdy)}_{\beta}(C) = Area(C)\Big(1 + \beta \ \mathrm{sign}\big(5 - \mathfrak{s}^{\Omega}_{\frac{1}{2},\frac{1}{2}} - \mathfrak{s}^{\Omega}_{1,1} - \mathfrak{s}^{\Omega}_{0,1} - \mathfrak{s}^{\Omega}_{1,0} - \mathfrak{s}^{\Omega}_{0,0}\big)\Big), \tag{7}$$

where $Area(C)$ denotes the area of cell C. In other words, the term $sign(\cdots)$ in (7) measures if all cell's corners and centers are out of Ω. Using this boundary principle function alone will refine nodes near the zero contour of $\mathfrak{s}^{\Omega}$ that is, of course, independent of f. It is not difficult to see that large β yields finer nodes near the boundary. Coupling the two principle functions, we arrive at the desired principle function in the following form

$$P_{\alpha,\beta}(C) = \begin{cases} P^{(int)}_{\alpha}(C) & \text{if } (x_i, y_j) \in \Omega \text{ for all } i, j \in \{0, \frac{1}{2}, 1\}, \\ P^{(bdy)}_{\beta}(C) & \text{otherwise.} \end{cases} \tag{8}$$

Under any circumstances, cell C is splittable if $Area(C) > \tau$ to ensure the desired resolution. If $Area(C) \leq \tau$ and if all $\mathfrak{s}^{\Omega}_{i,j} = -1$ ($i, j \in \{0, \frac{1}{2}, 1\}$), then cell C will be considered as *exterior* and will not be split further. The proposed ANR scheme may fail to capture boundary changes within an area smaller than τ.

In some situations, the function of interest f may in fact define well beyond Ω. For example, in the dual reciprocity method used by Chen et al. [31, 32, 33], rectangular domains enclosing the original (irregular) domain Ω are usually chosen for computation. Although one can simply employ the interior principle function (6), it makes sense to take Ω into consideration in order to avoid over-refinement outside of Ω. In this case, the level of boundary refinement (controlled by β) should very well depend on f also. Such a goal can be easily achieved by choosing $\beta = \beta(\alpha)$ adaptively in each cell such that $P^{(int)}_{\alpha}(C) = Area(C)(1 + \beta)$.

Figure 2 demonstrates four sets of nodes resulting from ANR scheme with principle function (8) and different values of α and β. Here, we pick the tolerances $\tau = 1/12^2$ for all runs. The parameter pair $[\alpha, \beta] = [0, 0]$ is the control run to give readers an intuition on how the parameters affect the ANR results. The next run, $[\alpha, \beta] = [1, 0]$ densely refines nodes around the peak of f. However, near the peak of f, the high density nodes resulting from $P^{(int)}_{\alpha}(C)$ are adjacent to very coarse cross-boundary nodes. This sudden change in point density arises due to the lack of communication between $P^{(int)}_{\alpha}(C)$ and $P^{(bdy)}_{\beta}(C)$. With $[\alpha, \beta] = [0, 100]$, the ANR scheme puts lots of nodes near boundary. The adaptive $\beta = \beta(\alpha)$, ANR results in a more uniform distribution and we see only two node-levels here. It is difficult to say which setting is superior over the others; the answer depends on what one wants to do. For example, if one is trying to measure maximum errors or capture delta-like functions, large α is desired. It is shown in [34] that an additional set of collocation equations near boundary is beneficial to the asymmetric RBF collocation method for solving PDEs. Or, if one is trying to track moving boundary as in [35], then large β for dense cross-boundary nodes is desired.

Carefully implemented data structure is the key to the algorithm's efficiency. To construct a Q-tree (or quadtree), we start with an initial cell containing Ω. This initial cell is denoted as the *root node* (or *tree-level* 0) of Q-tree . Starting from the root node, if cell C can be split into four subcells, i.e. $P_{\alpha,\beta}(C) > \tau$, the splitting process is repeated on each leaf until no leaf can be split again. This yields a Q-tree data structure. Out of boundary nodes identified by the signed function $\mathfrak{s}^{\Omega}$ will be removed. Keep the Q-tree in memory; it will become useful again later (see Section 3.2).

There are different ways to assign "levels" to the refined points. Here, we decide to go for a non-nested assignment. The smallest tree-level associated with cells' area less than τ will be called *cell-level 1*; cell-level then increases as we step up the Q-tree structure. The *node-level* of a given cell's corner (or refined node) is defined to be the cell-level of the smallest containing-cell.

Clearly, the proposed ANR scheme can generate sequential sets of collocation points X_N that getting dense quasi-uniformly as required by Theorem 1. We can see, also from Figure 2, that the quasi-uniform data sites required in Theorem 4 *cannot* be achieved by increasing α nor increasing β alone; it is obtained only by the tolerance $\tau \searrow 0$ with all the other parameters fixed. Letting the tolerance $\tau \searrow 0$ ensures that the maximum partition sizes $\max Area(\Omega_i) \searrow 0$ in Definition 1. The minimum partition sizes depends on both f and Ω. When they are both fixed (so as α and β in the principle function), it is straightforward to find a bound for $\min Area(\Omega_i)$ in terms of τ.

3. Hybrid algorithm

Given a function f on Ω, one can generate a set of N data sites, denoted by X_N. Simply by looking at the distributions in Figure 2, experienced RBF users can already see the underneath problems of ill-conditioning. Moreover, with minimal *a priori* knowledge about the function f, choosing basis functions is not obvious; for example, one may want to use GRBF for smooth functions but Wendland's CSRBF for the others. This motivates us to construct an algorithm that can approximate a wide variety of functions f up to reasonable accuracy. The general idea of such two-stage meshless approach date back to Schaback's 1995 article [36].

The algorithm runs on the data locations obtained by the ANR procedure presented in Section 2. We called the algorithm *hybrid* because both GRBFs and CSRBFs are used. Before discussing each building block of the algorithm in details, we first specify the notations used in Theorem 3 and in this section: the set of collocation points X_N is the set of N data sites output from the ANR scheme, the set of GRBF centers X_{n_1} is a subset of X_N selected by an adaptive greedy algorithm for a given GRBF kernel Φ_1, the set of CSRBF centers X_{n_2} is another subset of X_N used as centers of CSRBF kernel Φ_2, and the native spaces $\mathcal{U}_{n_1}, \mathcal{U}_{n_2}$ are the RBF trial subspaces, respectively, associated with the sets of centers X_{n_1}, X_{n_2} and kernels Φ_1, Φ_2 that are obtained by (4) with Ω replaced by the set of RBF centers.

3.1. Stage I: Global RBFs

To overcome the problem of ill-conditioning due to the non-uniform node distributions in X_N, we adopt the adaptive greedy technique to select good RBF centers (or $\mathcal{U}_{n_1}$) on-the-fly. The basic rationale is as follows: we adaptively pick from a large set of N initial input candidates $X_N = \{(x_i, y_i)\}_{i=1}^{N}$ and select a subset of n_1 RBF centers $X_{n_1} = \{(x_i, y_i)\}_{i=1}^{n_1}$ most likely with $n_1 \leq N$. There are different variants of adaptive greedy algorithms; for instance, see [37, 38, 39, 40, 41]. Here, we are going to employ the improved primal/dual-residual version found in the recent article [40]. The basic idea of this adaptive greedy algorithm is to make sequential selections of points-pair based on the largest entry-wise residuals. An overview is given below.

The number of input candidates (i.e. output from the ANR scheme) is usually large with no *a priori* control. The large and dense *original* GRBF interpolation matrix $\mathcal{A}$ is of size $N \times N$. The minimum norm solution, that is also the unique solution, can be obtained by the following constrained minimization problem:

$$\text{Minimize } \frac{1}{2}\mathbf{d}^T\mathcal{I}\mathbf{d}, \text{ subject to } \mathcal{A}\mathbf{d} - \mathbf{b} = 0. \tag{9}$$

Using the method of Lagrange multipliers, the primal/dual formulation of Problem (9) can be rewritten as

$$\begin{bmatrix} \mathcal{I} & \mathcal{A}^T \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}, \tag{10}$$

where $\mathbf{d} \in \mathbb{R}^N$ is the vector of unknown coefficients, $\mathbf{e} \in \mathbb{R}^N$ is the vector of Lagrange multipliers, $\mathcal{I} \in \mathbb{R}^{N\times N}$ is the unit matrix and $0 \in \mathbb{R}^{N\times N}$. Since any RBF interpolation matrix $\mathcal{A}$ can be completely determined by the data locations and the RBF centers, we denote any sub-matrix of $\mathcal{A}$ by $\mathcal{A}(\cdot,\cdot) : (\mathbb{R}^2)^N \times (\mathbb{R}^2)^N \longrightarrow \mathbb{R}^{N\times N}$, as a matrix function taking sets of data locations and sets of RBF centers, respectively, as the first and the second input arguments. Similarly, $\mathbf{b} : (\mathbb{R}^2)^N \longrightarrow \mathbb{R}^N$ can be seen as a vector function taking sets of data locations as input.

The *adaptive greedy algorithm* in [40] is built on (10). After step–k ($k \le N$), suppose the adaptive algorithm selects the sets $P_k = \{p_{(1)}, \cdots, p_{(k)}\} \subset X_N$ of data locations and $Q_k = \{q_{(1)}, \cdots, q_{(k)}\} \subset X_N$ of RBF centers ($p_{(\cdot)}$, $q_{(\cdot)} \in \mathbb{R}^2$). The next iteration will select the $(k+1)^{th}$ pair of collocation point and RBF center from the following subproblem

$$\begin{cases} \mathcal{A}_{(k)}\mathbf{d}^{(k)} = \mathbf{b}^{(k)}, \\ \mathcal{A}_{(k)}^T\mathbf{e}^{(k)} = -\mathbf{d}^{(k)}, \end{cases} \tag{11}$$

where $\mathcal{A}_{(k)} := Area(P_k, Q_k)$ and $\mathbf{b} := b(P_k)$. After solving (11) for $\mathbf{d}^{(k)}$ and $\mathbf{e}^{(k)}$, the primal–residual in $\mathbb{R}^N$ can be computed by

$$r^{(k)} = \mathcal{A}(X_N, Q_k)\mathbf{d}^{(k)} - \mathbf{b}. \tag{12}$$

Let $\mathbf{E}_{k\to N}\mathbf{d}^{(k)} \in \mathbb{R}^N$ be the extension of $\mathbf{d}^{(k)}$ by patching zeros into entries associated with the non-selected RBF centers. The dual–residual in $\mathbb{R}^N$ is given by

$$q^{(k)} = \mathbf{E}_{k\to N}\mathbf{d}^{(k)} + \mathcal{A}(P_k, X_N)^T\mathbf{e}^{(k)}. \tag{13}$$

The $(k+1)^{st}$ data locations $p_{(k+1)}$ and RBF centers $q_{(k+1)}$ can now be *greedily* selected from (12) and (13), respectively. This point-pair is then put into P_k and Q_k to form two bigger sets P_{k+1} and Q_{k+1}.

The iterations stop if either residual (12) or (13) is smaller than some tolerances, or if severe ill-conditioning in (11) is detected. The adaptive greedy algorithm is said to be *matrix-free* because evaluating (12) and (13) only requires submatrices of $\mathcal{A}$. The original matrix $\mathcal{A}$ will not be fully evaluated nor stored. Assuming the adaptive greedy algorithm terminates in $n_1 = \mathcal{O}(1)$ iterations, then the total work required is $\mathcal{O}(n_1^2(n_1 + 2N)) = \mathcal{O}(N)$ and the total storage requirement is $\mathcal{O}(n_1 N) = \mathcal{O}(N)$. Readers can refer to [40] and references therein for more implementation details.

3.1.1. Weights of collocations Although the convergence results in Section 1 hold for the standard least-squares approach (asymptotically), it makes sense to apply some appropriate collocation-weights to improve the performance in each discrete run. More importantly, the collocation-weights are introduced to avoid over-emphasizing the region of rapid variation in the collocation system.

In the first stage of the hybrid algorithm, we make use of the adaptive greedy algorithm to choose the set of n_1 RBF centers, denoted by X_{n_1}, for the global RBF approximation. The Stage I approximation is defined by

$$s_1(x,y) = \sum_{(x_i,y_i)\in X_{n_1}} d_i\, \Phi_{1,i}(x,y) \text{ such that } \mathbf{d} := \arg\min_{\mathbf{d}\in\mathbb{R}^{n_1}} \|f - s_1\|_{\ell^2(X_N;W_{X_N})},$$

where $W_{X_N} = \{w_i\}_{i=1}^N$ denotes the level-dependent collocation weights. The weighted $\ell^2(X_N)$-norm $\|\cdot\|_{\ell^2(X_N;W_{X_N})}$ means that the i-th collocation condition in $\|\cdot\|_{\ell^2(X_N)}$ is multiplied by the weight w_i $(i = 1,\ldots,N)$.

In Figure 3, we consider a function f consisting a smooth exponential background and a spike. The ANR scheme is run with $\alpha = 1$, $\beta = 0$, and $\tau = 5^{-2}$. The reason we choose the non-nested levels assignment will now be made clear. From the *node-level*, we can see that the data sites under the spike are between node-level 3 to 6. If, and this is a big-IF, one can look at the plot of f and decide to remove the spike from the collocation system, then this can be done simply by assigning zero-weights to collocation conditions in node-levels 3 to 6. The adaptive greedy algorithm is shown to be highly efficient in approximating smooth function; the approximation s_1 shown in Figure 3 is obtained by the MQ basis as Φ_1 with a constant shape parameter $c = 1$. The residual $f - s_1$ contains only the spike now. The residual will be taken care of in Stage II of the hybrid algorithm with CSRBF.

If we want the proposed hybrid algorithm to become a care-free blackbox, the look-and-decide approach we just used is not feasible. Solely depending on the node-levels to zero out some collocation conditions will lead to unsatisfactory results in general. For example, applying the ANR scheme with exactly the same setting to the Matlab's *PEAKS* functions yields five levels of data locations. Zeroing-out all points in node-levels 3 to 5 will disable 98.8% of the collocation conditions. Clearly this is bad and we omit the numerical demonstrations. Of course, point density or other relevant means can be taken into account; the "zero-out" idea may still be useful (Figure 3 is one of the examples). For the sake of this work, however, we decide to take a more neutral approach.

Due to the Q-tree structure and the non-nested levels assignment, the spacings between ANR outputs are very systematic and predictable. Take Figure 3 as an example again. The minimum separating distances of points in node-level 1 is $h = 2^{-3}$ meaning that they are corners of some leaf-cells with area h^2. The minimum separating distances of points in node-level 2 is $h/2$; that of node-level 3 is $h/4$ and so on. If all collocation conditions are treated equally, then the number of imposed conditions per unit area is much higher in region with high data density; i.e., under the spike of Figure 3. Following the same logic, we impose a level-dependent weight: $w_\ell = 2^{2(1-\ell)}$ for all node-level ℓ in X_N. In this case, there is no bias towards different levels of data locations and the advantages of the adaptive greedy algorithm on smooth functions are well preserved.

3.2. Stage II: CSRBFs

To initialize Stage II of the hybrid algorithm, we use the Stage I GRBF approximation s_1 to define the residual function $f - s_1$. With a user-defined residual tolerance $\gamma > 0$, the set of CSRBF centers is defined as

$$X_{n_2} := \big\{(x, y) \in X_N \, : \, |(f - s_1)(x, y)| \geq \gamma\big\}.$$

In practice, the value of γ should be set to a specific application-dependent (acceptable) accuracy-level. Intuitively, if Stage I cannot complete the job within the desired accuracy given by γ, then Stage II will be activated for further residual reduction.

In general, we are going to see $n_1 \ll n_2$ unless the function of interest is so smooth that it can be well approximated solely by the global RBFs in Stage I. The well-known advantage of using CSRBF is that the collocation matrix can be made sparse by scaling the support radius. Let Φ_2 be the selected CSRBF kernel. The Stage II approximation is defined as

$$s_2(x, y) = \sum_{(x_i, y_i) \in X_{n_2}} \widetilde{d}_i \, \Phi_{2,i}(x, y) \text{ such that } \widetilde{\mathbf{d}} := \arg \min_{\mathbf{d} \in \mathbb{R}^{n_2}} \|s_2 - (f - s_1)\|_{\ell^2(X_N)},$$

where the i-th basis function with support radius ρ_i is given as

$$\Phi_{2,i}(x, y) = \Phi_2 \left(\frac{1}{\rho_i} \big\|(x, y) - (x_i - y_i)\big\|_2 \right), \quad \text{for} \quad (x_i, y_i) \in X_{n_2}.$$

Recall the observations from Figure 3, the distribution in X_{n_2} is expected to be highly non-uniform. Most likely, data sites in X_{n_2} are dense in the region where f changes rapidly. This hints us that the support radius ρ_i should not be too large. Using the inherent tree structure, the support radius ρ_i for each CSRBF center $(x_i, y_i) \in X_{n_2}$ can be estimated providing the desired "bandwidth" of the collocation matrix. Such spatially variable shape parameters have been used successfully; interested readers can find more in the articles [28, 42, 43] and the references in the book [14].

Using the nearest neighbors search can avoid computation of the already-known zero entries in the sparse collocation matrix (and also for efficiently evaluating s_2). This is essential for efficient implementation of the CSRBF scheme. Although most researchers use kd-trees (downloadable from the MATLAB CENTRAL File Exchange), in our situation, the Q-tree is already available from the ANR computation; it is a byproduct of the ANR scheme. When the support size is relatively small compared to the number of points in X_N, both Q-tree and kd-tree are equally efficient (see for instance [44]). Moreover, with very minor tweaks, the search of nearest neighbors and the search of support radius (providing the desired number of points to be covered) can be done simultaneously. Implementing Stage II of the hybrid algorithm should be straightforward due to numerous references available in the recent book by [14]. However, assigning support radius solely by the number of covered data sites may yield oversized basis near boundary. Suppose $(x_i, y_i) \in X_{n_2}$ is in level-ℓ whose corresponding leaf-cell area is h_i^2. If we set $\rho_i = kh_i$, then the CSRBF $\Phi_{2,i}$ will cover about $\frac{1}{4}\pi(2k + 1)^2$ points in X_N; this estimation becomes more accurate in higher node-levels.

4. Numerical examples

The proposed hybrid algorithm, contains three main routines: the ANR scheme for finding data sites X_N for collocation, Stage I for finding centers of GRBF and approximation s_1, and

Stage II for finding support radius of CSRBF and approximation s_2. The proposed algorithm will be applied to approximate various functions. Numerical errors are measured in the relative means with respect to the $L^2(\Omega)$–norm:

$$\mathcal{E}_2 := \frac{\|f-(s_1+s_2)\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}}.$$

The maximum errors are sometimes used for comparison with previous literatures. Before running the hybrid algorithm, the following parameters' values are suggested. For $\Omega \subseteq [-1,1]^2$ (scale if necessary),

- **Principle tolerance:** Take $\tau = 2^{-k}$ $(k > 0)$, reduce if necessary, to yield around $N = 1000$ to 2000 data sites,
- **Interior principle parameter:** $0 \le \alpha \le 0.5$ to avoid over-concentrated data sites; that is, the term $\int_C |\nabla f|\, dx$ is disabled in the principle,
- **Boundary principle parameter:** To obtain quasi-uniform data sites, select $\beta = \beta(\alpha)$ adaptively such that $P_\alpha^{(int)}(C) = Area(C)(1+\beta(\alpha))$;
- **Residual tolerance:** $\gamma = 10^{-5}$; residuals at data sites larger than γ will be past from Stage I to II,
- **GRBF:** $\Phi_1(r) = \sqrt{1+r^2}$, i.e. the multiquadrics basis with shape parameter $0.1 \le c \le 1$.
- **Collocation weights:** $w_\ell = 2^{2(1-\ell)}$ for collocations at data sites in node-level ℓ,
- **CSRBF:** $\Phi_2(r) = (35r^2+18r+3)\max(0,(1-r))^6$, i.e. the Wendland's C^4–basis,
- **Support radius:** ρ_i such that each *interior* CSRBF Φ_2 covers about 100 points in X_N; in our implementation, we take $\rho_i = 5h_i$.

All numerical demonstrations in this section make use of the same set of parameters with either $(c, \alpha) = (1.0, 0.0)$ or $(0.1, 0.5)$. To avoid overrunning, some examples only come with short discussions. We provide results of surface approximation to six different types of functions and of Burgers' equation. This, hopefully, will provide more confidences to use the proposed algorithm as blackbox. The conclusion we are going to draw is as follows: $(c, \alpha) = (1.0, 0.0)$ is suitable for most function approximations whereas $(c, \alpha) = (0.1, 0.5)$—using larger n_1—is not as efficient but it will be shown stable enough for solving time-dependent PDEs.

4.1. Example: Smooth–Octopus

In the first example, we justify the selection of MQ shape parameter $c = 1$ and interior principle parameter $\alpha = 0$. To show that the hybrid algorithm works ideally on smooth functions in this setting, we consider the following tilted exponential function

$$f(x,y) = \frac{x+y}{10} + \exp\big(10(-(x+0.1)^2 - y^2)\big),$$

on an asterisk-shape domain given in polar coordinates as

$$r(\theta) = 1 + \cos(4\theta)^2, \quad \theta \in [0, 2\pi]$$

to make a *smooth-octopus*. We let the principle tolerance τ go down as k^{-2} $(k = 3, 4, \ldots)$. Stage II is activated only in cases when $c = 1.5$ and $c = 2$; the other runs (with smaller c) completed successfully after Stage I with residual less than the residual parameter γ. It has been previously reported that, although the adaptive greedy algorithm can prevent numerical instability caused by large shape parameters, a reasonable c value can still be beneficial to the accuracy.

In Figure 4 *(Left/Lines)*, the RMS errors associated with various values of c are shown in lines verse the number of data sites N (i.e., ANR output refined points). We can clearly see the improvement in accuracy when c increases from 0.1 to 1.0. Larger shape parameters (i.e. $c = 1.5$ and 2.0) result in relatively poor accuracies. Note, the numbers of input collocation points are not the most important factor here. Instead, the degrees of freedom (DoF) or numbers of basis used in numerical expansions $s_1 + s_2$ are directly related to the speed of the algorithm. The DoF for $c = 1.5$ and $c = 2.0$ are large due to the introduction of CSRBF in s_2; whereas, all DoF in the case of $c = 0.1$ are due to s_1. In Figure 4 *(Left/Markers)*, we show the RMS errors against the degrees of freedom. In terms of both accuracy and efficiency, using shape parameter $c = 1$ seems to be better than the others values of c (in here and a few other unreported tests) for approximating smooth functions.

Running a similar test with various interior principle parameters, we can see in Figure 4 *(Right)* that $\alpha = 0$ is superior over other choices when approximating smooth function in terms of computational cost; that is, achieving the same accuracy with smaller N (see lines in the figure) and in terms of the DoF (see markers).

Lastly, in Figure 5, a numerical approximation and the associated residual function obtained by the proposed Hybrid algorithm (with $c = 1$ and $\alpha = 0$) are shown as demonstration. The error distribution is fairly uniform except at the tips of the octopus' arms. We see that some interior errors (on the body of the octopus) and the maximum errors are in the same order of magnitude. In this example, the boundary refinement in ANR is satisfactory.

4.2. Example: PEAKS

In the second example, the hybrid algorithm is used to approximate the Matlab's PEAKS function, another smooth function. Since the function contains some flat regions, from the design of the ANR scheme, we know that the coarse grids will not be refined until the principle tolerance τ is reduced to the next power of 2.

The relative error profiles of both $(c, \alpha) = (1.0, 0.0)$ and $(0.1, 0.5)$ are shown in Figure 6. The numerical approximation in Figure 6 *(Left)*, for the former setting, may appear that the error along the boundary is relatively high. However, this is related to the residual tolerance γ. Reducing γ can, with no doubt, tag more data sites to be passed onto Stage II. Better accuracy can be obtained with more work in the second stage. Accuracies of the proposed method clearly outbid the results shown in the paper by [38] using a determinant–based adaptive greedy algorithm. In terms of convergent behaviors, Figure 6 *(Right)*—especially for $(c, \alpha) = (0.1, 0.5)$—is more monotonic and less oscillatory compared to the results reported in [40].

4.3. Example: (Elongated) Near-Singularities

We now turn our focus to functions with rapid variations and further investigate the numerical performances of the suggested parameters (c, α). For the sake of easy comparison to literatures, we only consider square domains hereafter. In this example, we consider a standard test function with multiple regions of rapid variations for adaptive methods found in [45, 46]

$$f(x,y) = 1 + \exp\big(-((x')^2+(y')^2)/2\mu_1\big) + \exp\big(-((x'')^2/2\mu_2) - ((y'')^2/2\mu_3)\big),$$

where $\mu_1 = \mu_2 = 10^{-3}$, $\mu_3 = 10^{-1}$, $x' = x - 0.5$, $y' = y - 0.5$, $[x'', y'']^T = \mathbf{R}\,[x+0.1, y+0.1]^T$ with rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos\pi/3 & \sin\pi/3 \\ -\sin\pi/3 & \cos\pi/3 \end{bmatrix}.$$

In Figure 7, the numerical approximations and point distributions for three different values of principle tolerances τ are shown for the case $(c,\,\alpha) = (1.0,\,0.0)$.

From $\tau = 1/6^2$ to $\tau = 1/10^2$, the coarse grids are refined and we can see a significant improvement in accuracy from $\mathcal{E}_2 = 9.01E-3$ with $N = 484$ down to $\mathcal{E}_2 = 1.31E-3$ with $N = 1543$. From $\tau = 1/10^2$ to $\tau = 1/14^2$, refinements are added to the near-singularities only. The relative error further drops to $\mathcal{E}_2 = 9.34E-4$ with $N = 1902$. In all three shown cases, the numbers of GRBF used are very similar: 204, 252, and 240, respectively, for $\tau = 1/6^2$, $1/10^2$, and $1/14^2$. The main difference is the numbers CSRBF used: they are, respectively, 483, 1540, and 1899. The observed accuracy here, in terms of maximum error, is about one to two orders of magnitude better than the variable shape parameters MQ approach employed by [45]. Note that our approach only uses full matrices with about 200 columns in Stage I and sparse matrices in Stage II in contrast to the $N \times N$ full matrices encountered by Libre *et al.* Even though the ANR scheme locally refines data sites around the rapid variations, we can see in Figure 8 that the number of nonzero entries in the Stage II collocation matrices only increases linearly with the number of data sites.

Figure 9 shows a numerical approximation and convergent profiles. Using $(c,\,\alpha) = (1.0,\,0.0)$, the rounded spike is very well approximated in Figure 9 *(Left)*. Errors near the elongated spike are slightly higher—mostly concentrated near the valley of the spike. Similar to the PEAKS function, the test function here is completely flat near the boundary. We observe that the maximum error occurs again near the boundary; more specifically, it is at $(x,y) = (1,1)$. The oscillatory convergent pattern for $(c,\,\alpha) = (1.0,\,0.0)$ in Figure 9 *(Right)* is quite obvious here, but present in other tests too. For $(c,\,\alpha) = (0.1,\,0.5)$, the convergent profile starts from larger N indicating that the number of data sites is larger in this case. It is solely due to the way ANR works: as briefly mentioned before, coarse cells in the flat region are refined only when the principle tolerance τ gets smaller than the *next* power of 2. As τ decreases in between two powers of 2, coarse cells stay while numbers of fine cells increases. Although the collocation-weights are designed to minimize the effect, the accuracy of Stage I seems to be affected by the unbalance of coarse/fine cells (or data sites) distributions. Hence, in the suggested parameters set, we suggest taking $\tau = 2^{-k}$, $k > 0$.

4.4. Example: Boundary-Layers

To make further comparison to the variable shape parameters MQ approach, we consider another example of theirs,

$$f(x,y) = \exp\big(-50(x+1)^2\big) + \exp\big(-50(y+1)^2\big).$$

Convergent results can be found in Figure 10. For $(c,\,\alpha) = (1.0,\,0.0)$, it takes $N = 501$ data sites to obtain better a maximum error than that reported by [45] with $N = 1201$ yielding a maximum error of 6.21E-2. In this setting, the hybrid algorithm results in accuracy a magnitude smaller than those reported in [45]. When we apply the less efficient setting $(c,\,\alpha) = (0.1,\,0.5)$, another magnitude of improvement can be achieved.

There are many factors that cause the higher errors along the lines $x = -1$ and $y = -1$. These lines, respectively, are the turning points of the exponential functions in f that are locally flat. The employed ANR scheme, without the curvature term (i.e. $\alpha = 0$), is unable to detect the high curvatures at the turning points. The desire of refining these appealingly flat regions is low. Looking closely at the ANR refinement process, one can see that data sites are refined one-cell away from the boundary when $\alpha = 0$. This is the limitation of the current blackbox; enforcing too many data sites near boundary or re-activating the curvature in the interior principle function just for this test (with $\alpha = 0.5$) will reduce efficiency in other cases. If, however, we know beforehand that we are dealing with a boundary layer problem, then there are loads of tweaks one can apply to improve accuracy here.

4.5. Example: Shock-Formation

We consider the C^0–function

$$f(x,y) = \exp\big(1 - 20\max(0.04, x' - y' - 0.4)(x' - y' - 0.4)^2 - 4(x' + y')^2\big),$$

where $x' = \sqrt{2}x$ and $y' = \sqrt{2}y$. The shape of f can be commonly seen in articles solving Burgers' equations or other types of nonlinear transport. To avoid giving advantages to the ANR scheme's tensor product structure, rotation is applied so that the shock front does not align with either the x– or y–axis. Numerical results are shown in Figure 11. Even though f clearly lies outside the native spaces of the MQ and the CSRBF (despite the fact that we use multi-supports), the convergence and accuracy here are better than the C^∞ case in the previous examples. Note also that the advantages of using $(c,\,\alpha) = (0.1,\,0.5)$ are clearly shown in Figure 11 (*Right*).

4.6. Example: Moving-Front

Our last example on function approximation, taken from the book by Fasshauer [14], mimics the heavystep function or situations seen in moving front. Consider the function given as

$$f(x,y) = \frac{\tanh(9(y-x)) + 1}{\tanh(9) + 1}.$$

If we compare the results here with the those reported in [14] for the Hermite interpolation, Lagrange interpolation and clustered Lagrange approaches, it is obvious that Hermite interpolation is still the most accurate method for this example. But our results are comparable with the other Lagrange-based methods.

If we turn our focus to the error distribution, the maximum errors of all tested methods in [14] occur somewhere at the moving front. In Figure 12, however, we see that the maximum error of the hybrid algorithm is at the flat region near the boundary. This further confirms that our proposed method has slight bias towards the interior and regions of rapid variation.

4.7. Application to Burgers' Equation

Up to this point, we observe the pros and cons for the suggested $(c,\,\alpha)$; that is, using $(1.0,\,0.0)$ is highly efficient and yields good approximation for smooth functions, whereas using $(0.1,\,0.5)$

gives better accuracy on nearly-singular functions. In this demonstration, we consider the following Burgers' equation:

$$\frac{\partial u}{\partial t} + \triangle\left(\frac{1}{2}u^2\right) = 0, \quad t \in \mathbb{R}^+,\ (x,y) \in [-1,1]^2; \tag{14}$$

with $x' = x + 0.4$ and $y' = y + 0.4$, the nonsmooth initial condition is given as

$$u(x,y,0) = \begin{cases} \exp \dfrac{(x')^2 + (y')^2}{(x')^2 + (y')^2 - 0.25}, & (x')^2 + (y')^2 < 0.25, \\ 0, & \text{Otherwise.} \end{cases}$$

Our numerical treatment is as follows: the backward Euler method is applied to discretize (14). Next, we linearize and obtain a nonlinear modified Helmholtz equation. To adaptive place collocation points in each time update, the proposed node refinement scheme is applied to the numerical solution u^n. This set of points will be used for solving u^{n+1} that is done in two-stage—similar to what we discuss in the previous chapter with minor modifications.

As the solution evolves in time, we see a shock formation whose profile is very similar to the problem of approximation shown in Example 4.5. Although the setting $(c,\ \alpha) = (1.0,\ 0.0)$ yields reasonable accuracy in a single-run for surface approximation, it turns out to be unstable as we iterate over time. In Figure 13, numerical results obtained by parameters $(c,\ \alpha) = (0.1,\ 0.5)$ and time step $dt = 0.02$ are shown at different time $t = 0.12,\ 0.48,\ 0.84,\ 1.2$. On the right, the data sites are shown with contour lines of the numerical solutions. We can see that the number sites remains relatively small over all the snapshots. The maximum norms of our numerical solutions show nice agreement with that reported in [47]. Other meshless or localized-meshless approaches for the problem are discussed in [48, 49, 50, 51, 52]; some of them require larger sets of collocation points and relatively small time step (around $dt = 0.005$).

5. Conclusion

A hybrid algorithm for meshfree approximation, with mixed global and compactly supported radial basis functions along with different available techniques, is proposed. An adaptive node refinement scheme, which is both function– and domain–dependent, is proposed to generate data sites for collocation. The primal/dual-residual adaptive greedy algorithm is employed for trial subspace selection that circumvent the problem of ill-conditioning in the solution process of global radial basis function. Constant shape parameter used in global radial basis functions ensure efficiency when one is dealing with smooth functions, whereas variable support radius are used in compactly supported radial basis functions to ensure that for the less-smooth. A wide range of functions, including smooth exponential, PEAKS, near-singular, boundary-layer, shock-formation, and moving-front functions are studied either on irregular or rectangular domain. By a fixed set of suggested parameters' values ($c = 1,\ \alpha = 0$), a balance is made for the hybrid algorithm to work on all these functions/domains of different types. The proposed algorithm is also applied to solve shock-forming Burgers' equations as a demonstration to its application to PDEs.

From our numerical results, we believe there is still a lot of room for future work to improve the strategy for picking the boundary principle parameter β. For example, with the extra computational cost, we could re-run ANR on the residual functions or run ANR after running

Stage I on regular grids. Simpler and more cost-efficient options include: use $P_\alpha^{(int)}(C) + P_\beta^{(bdy)}(C)$ (as boundary principle) or pick $\beta(\alpha)$ so that $P_\alpha^{(int)}(C) = \epsilon Area(C)(1+\beta),\ 0 < \epsilon < 1$ to refine along the whole boundary, or use $\max\left(P_\alpha^{(int)}(C),\, P_\beta^{(bdy)}(C)\right)$ to ensure certain boundary sites density. All of the above can artificially enforce *finer* boundary refinement without a huge overhead to the computational cost. Moreover, we sometimes observe that the refinements of coarse cells lack-behind those of the fine cells. When this happens, maximum errors are usually in smooth region. The extra computation costs in handling more "fine-level" data sites is therefore spent in vein. Instead of disabling the curvature terms in the principle functions, we can impose an upper-bound to the node-levels. This will serve the same purpose to prevent over-concentrated data sites. Using smaller collocation-weights in higher node-levels could also be a solution. At the same time, function information can be better captured.

With more fine tuning of user-parameters for specific-types of functions, the proposed method will always yield better results than what is shown in this paper. However, running the hybrid algorithm with our suggested user-parameters as a blackbox algorithm is generous and robust enough to handle a wide range of functions.

REFERENCES

1. T. O. Kwok, L. Ling, On convergence of a least-squares Kansa's method for the modified Helmholtz equations, Adv. Appl. Math. Mech. 1 (3) (2009) 367–382.
2. R. Schaback, Error estimates and condition numbers for radial basis function interpolation, Adv. Comput. Math. 3 (3) (1995) 251–264.
3. R. Schaback, Approximation by radial basis functions with finitely many centers, Constr. Approx. 12 (3) (1996) 331–340.
4. H. Wendland, Sobolev-type error estimates for interpolation by radial basis functions, in: Surface fitting and multiresolution methods (Chamonix–Mont-Blanc, 1996), Vanderbilt Univ. Press, Nashville, TN, 1997, pp. 337–344.
5. W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, Comput. Math. Appl. 24 (12) (1992) 121–138.
6. W. R. Madych, S. A. Nelson, Multivariate interpolation and conditionally positive definite functions, Approx. Theory Appl. 4 (4) (1988) 77–89.
7. J. Yoon, Spectral approximation orders of radial basis function interpolation on the Sobolev space, SIAM J. Math. Anal. 33 (4) (2001) 946–958.
8. F. J. Narcowich, J. D. Ward, H. Wendland, Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting, Math. Comp. 74 (250) (2005) 743–763.
9. M. Buhmann, Convergence of univariate quasi-interpolation using multiquadrics., IMA J. Numer. Anal. 8 (3) (1988) 365–383.
10. M. D. Buhmann, S. Dinew, E. Larsson, A note on radial basis function interpolant limits, IMA J. Numer. Anal., published online 30 (2) (2009) 543–554.
11. I. Jackson, An order of convergence for some radial basis functions., IMA J. Numer. Anal. 9 (4) (1989) 567–587.
12. Z. M. Wu, R. Schaback, Local error estimates for radial basis function interpolation of scattered data, IMA J. Numer. Anal. 13 (1) (1993) 13–27.
13. M. D. Buhmann, Radial basis functions: theory and implementations, Vol. 12 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.
14. G. E. Fasshauer, Meshfree approximation methods with Matlab., Interdisciplinary Mathematical Sciences 6. Hackensack, NJ: World Scientific., 2007.
15. H. Wendland, Scattered data approximation, Vol. 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.
16. C. F. Lee, L. Ling, R. Schaback, On convergent numerical algorithms for unsymmetric collocation, Adv. Comput. Math 30 (4) (2009) 339–354.
17. B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. 48 (5-6) (2004) 853–867.

18. L. Ling, E. J. Kansa, A least-squares preconditioner for radial basis functions collocation methods, Adv. Comput. Math 23 (1-2) (2005) 31–54.
19. L. Ling, E. J. Kansa, Preconditioning for radial basis functions with domain decomposition methods, Math. Compt. Modelling 40 (13) (2004) 1413–1427.
20. A. Karageorghis, C. S. Chen, Y. S. Smyrlis, A matrix decomposition method for solving large scale problems using radial basis functions - approximation of functions and their derivatives, Appl. Numer. Math. 57 (2007) 304–319.
21. K. Li, Q. B. Huang, J. L. Wang, L. G. Lin, An improved localized radial basis function meshless method for computational aeroacoustics, Eng. Anal. Bound. Elem. 35 (1) (2011) 47–55.
22. J.-S. Chen, W. Hu, H.-Y. Hu, Localized radial basis functions with partition of unity properties, in: Progress on meshless methods, Vol. 11 of Comput. Methods Appl. Sci., Springer, New York, 2009, pp. 37–56.
23. M. A. Golberg, C. S. Chen, H. Bowman, H. Power, Some comments on the use of radial basis functions in the dual reciprocity method, Comput. Mech. 21 (2) (1998) 141–148.
24. P. W. Partridge, B. Sensale, The method of fundamental solutions with dual reciprocity for diffusion and diffusion-convection using subdomains, Eng. Anal. Bound. Elem. 24 (9) (2000) 633–641.
25. C. S. Chen, A. S. Muleshkov, M. A. Golberg, The numerical evaluation of particular solutions for Poisson's equation—a revisit, in: Boundary elements, XXI (Oxford, 1999), Vol. 6 of Int. Ser. Adv. Bound. Elem., WIT Press, Southampton, 1999, pp. 313–322.
26. C. Chen, Y. F. Rashed, Evaluation of thin plate spline based particular solutions for Helmholtz-type operators for the DRM., Mech. Res. Commun. 25 (2) (1998) 195–201.
27. M. J. Johnson, A symmetric collocation method with fast evaluation, IMA J. Numer. Anal. 29 (3) (2009) 773–789.
28. T. A. Driscoll, A. R. H. Heryudono, Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, Comput. Math. Appl. 53 (6) (2007) 927–939.
29. R. Opfer, Multiscale kernels, Adv. Comput. Math. 25 (4) (2006) 357–380.
30. L. Ling, Finding numerical derivatives for unstructured and noisy data by multiscale kernels, SIAM J. Numer. Anal. 44 (4) (2006) 1780–1800.
31. C. S. Chen, C. A. Brebbia, H. Power, Dual reciprocity method using compactly supported radial basis functions, Comm. Numer. Methods Engrg. 15 (2) (1999) 137–150.
32. C. Chen, M. Golberg, R. Schaback, Recent developments in the dual reciprocity method using compactly supported radial basis functions, in: Y. Rashed, , C. Brebbia (Eds.), Transformation of Domain Effects to the Boundary, WIT Press, 2003, pp. 183–225.
33. M. A. Golberg, C. S. Chen (Eds.), The dual reciprocity method and radial basis functions, Elsevier, Amsterdam, 2000, eng. Anal. Boundary Elements **24** (2000), no. 7-8.
34. A. I. Fedoseyev, M. J. Friedman, E. J. Kansa, Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary, Comput. Math. Appl. 43 (3-5) (2002) 439–455.
35. X. Zhou, Y. Hon, K. Cheung, A grid-free, nonlinear shallow-water model with moving boundary, Eng. Anal. Bound. Elem. 28 (9) (2004) 1135–1147.
36. R. Schaback, Creating surfaces from scattered data using radial basis functions, in: Mathematical methods for curves and surfaces (Ulvik, 1994), Vanderbilt Univ. Press, Nashville, TN, 1995, pp. 477–496.
37. Y. C. Hon, R. Schaback, X. Zhou, An adaptive greedy algorithm for solving large RBF collocation problems, Numer. Algorithms 32 (1) (2003) 13–25.
38. L. Ling, R. Opfer, R. Schaback, Results on meshless collocation techniques, Eng. Anal. Bound. Elem. 30 (4) (2006) 247–253.
39. L. Ling, R. Schaback, Stable and convergent unsymmetric meshless collocation methods, SIAM J. Numer. Anal. 46 (3) (2008) 1097–1115.
40. L. Ling, R. Schaback, An improved subspace selection algorithm for meshless collocation methods, Int. J. Numer. Methods Eng. 80 (13) (2009) 1623–1639.
41. R. Schaback, H. Wendland, Adaptive greedy techniques for approximate solution of large RBF systems, Numer. Algorithms 24 (3) (2000) 239–254.
42. N. Flyer, E. Lehto, Rotational transport on a sphere: local node refinement with radial basis functions, J. Comput. Phys. 229 (6) (2010) 1954–1969.
43. B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, Comput. Math. Appl. 54 (3) (2007) 379–398.
44. M. T. Goodrich, R. Tamassia, Data structures and algorithms in Java. 3rd ed., Hoboken, NJ: John Wiley &amp; Sons. xvii, 681 p. $ 91.95 , 2004.
45. N. A. Libre, A. Emdadi, E. J. Kansa, M. Shekarchi, M. Rahimian, A multiresolution prewavelet-based adaptive refinement scheme for RBF approximations of nearly singular problems, Eng. Anal. Bound. Elem. 33 (7) (2009) 901–914.
46. O. V. Vasilyev, N. K.-R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems,

J. Comput. Phys. 206 (2) (2005) 412–431.
47. J. Fürst, T. Sonar, On meshless collocation approximations of conservation laws: preliminary investigations on positive schemes and dissipation models, ZAMM Z. Angew. Math. Mech. 81 (6) (2001) 403–415.
48. J. Behrens, A. Iske, Grid-free adaptive semi-Lagrangian advection using radial basis functions, Comput. Math. Appl. 43 (3-5) (2002) 319–327.
49. G. Liu, D. Huynh, Y. Gu, An adaptive meshfree collocation method for static and dynamic nonlinear problems, Computational Methods (2006) 1459–1464.
50. B. J. Shi, D. W. Shu, S. J. Song, Y. M. Zhang, Solving Burgers' equation by a meshless method, Modern Phys. Letter B 19 (28-29) (2005) 1651–1654.
51. A. Hashemiana, H. M. Shodja, A meshless approach for solution of Burgers' equation, J. Comp. Appl. Math. 220 (1-2) (2008) 226–239.
52. B. Hosseini, R. Hashemi, Solution of Burgers' equation using a local-RBF meshless method, Int. J. Compt. Methods Eng. Sci. Mech. 12 (1) (2011) 44–58.

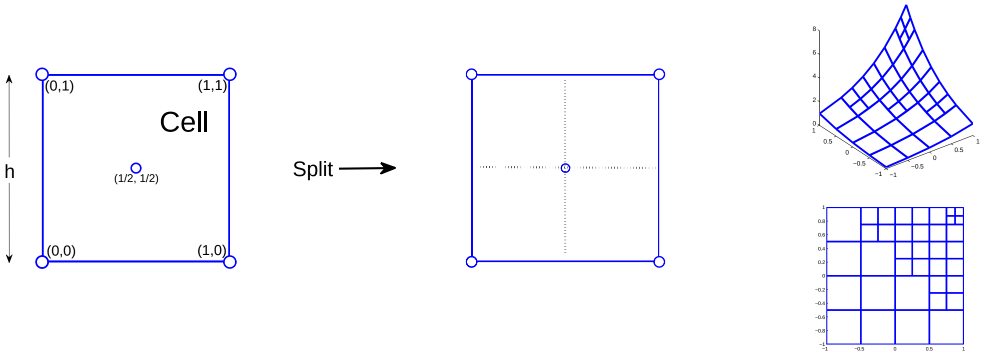

Figure 1. (Left) Generic labeling for each individual cell; split cell if the principle $P(C) > \tau$. (Right) Schematic illustration of cells splitting in 2D and 3D views.

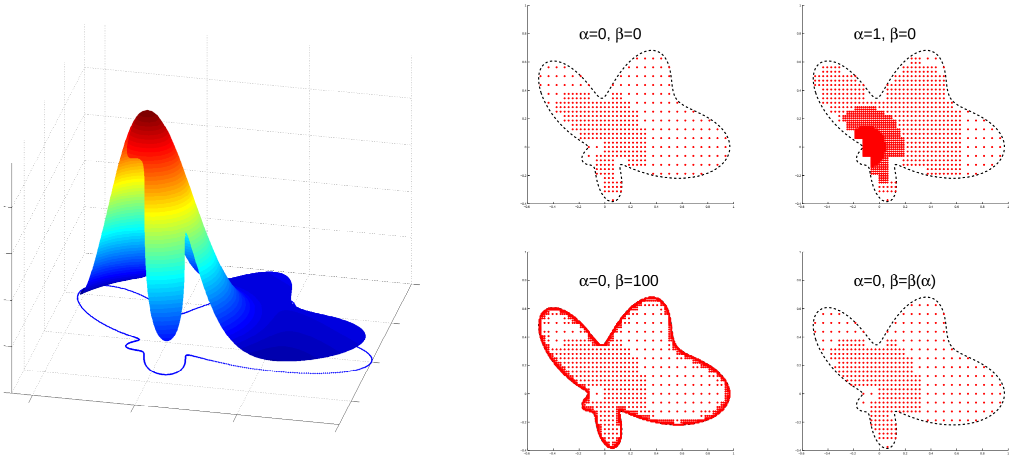

Figure 2. Effects of α and β in the principle function $P_{\alpha,\beta}$ when applying ANR to $f(x,y) = (x+y)/10 + \exp(10(-(x+0.1)^2 - y^2))$.

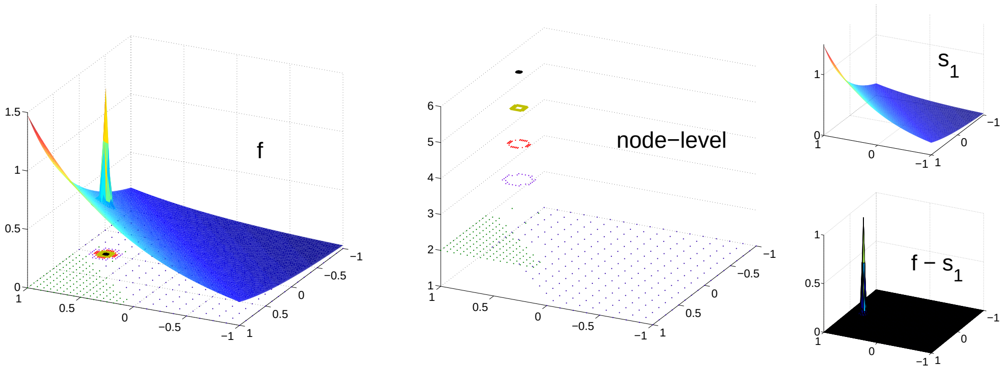

Figure 3. Non-nested level structures on the ANR output refined nodes. Approximation s_1 to $f = \exp(x+y)/5 + \exp(-1000((x-3/4)^2+y^2))$ obtained by assigned nonzero weights to the first two node-levels only.

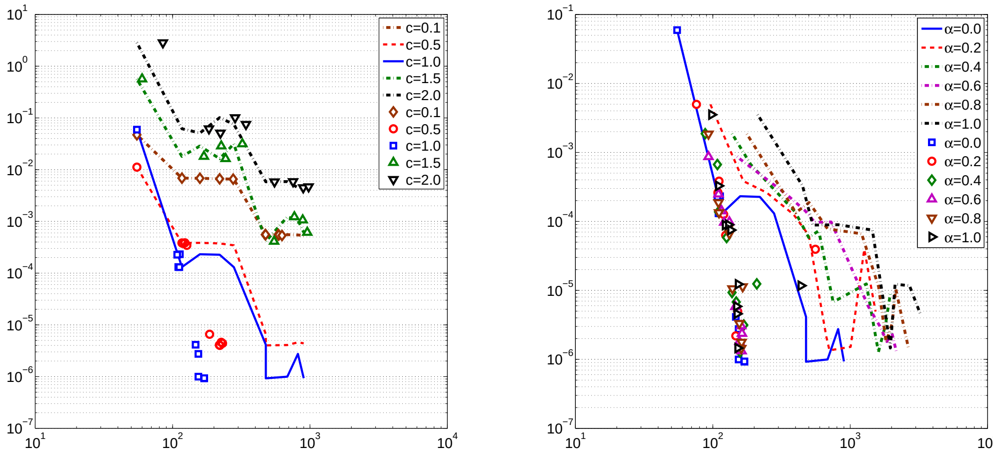

Figure 4. Example 4.1: RMS errors against different numbers of data cites and degrees of freedom in the numerical approximations, respectively, given in lines and markers. (Left) various MQ shape parameters c with fixed $\alpha = 0$; (Right) various interior principle parameters α with fixed $c = 1$.

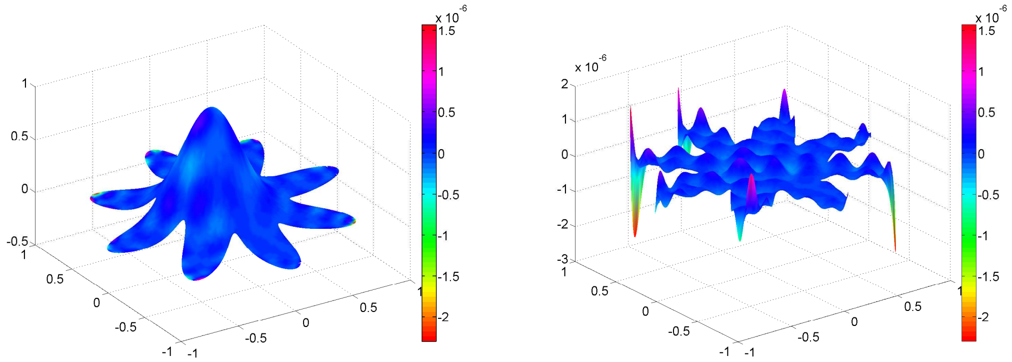

Figure 5. Example 4.1: (Left) Numerical approximation to the smooth-octopus and (Right) associated residual function.

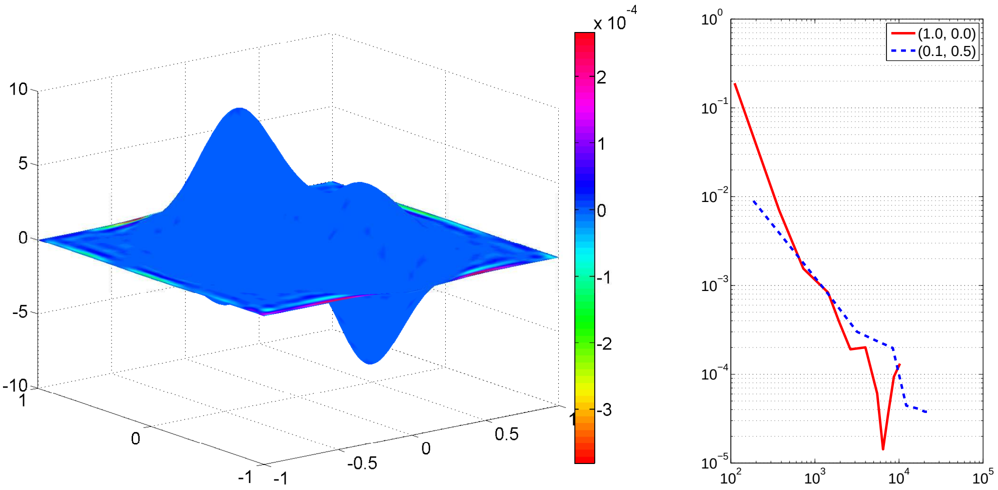

Figure 6. Example 4.2: (Left) a numerical approximation to the PEAKS function; (Right) relative $L^2(\Omega)$ errors $\mathcal{E}_2$ with different (c, α) against the number of data sites N.

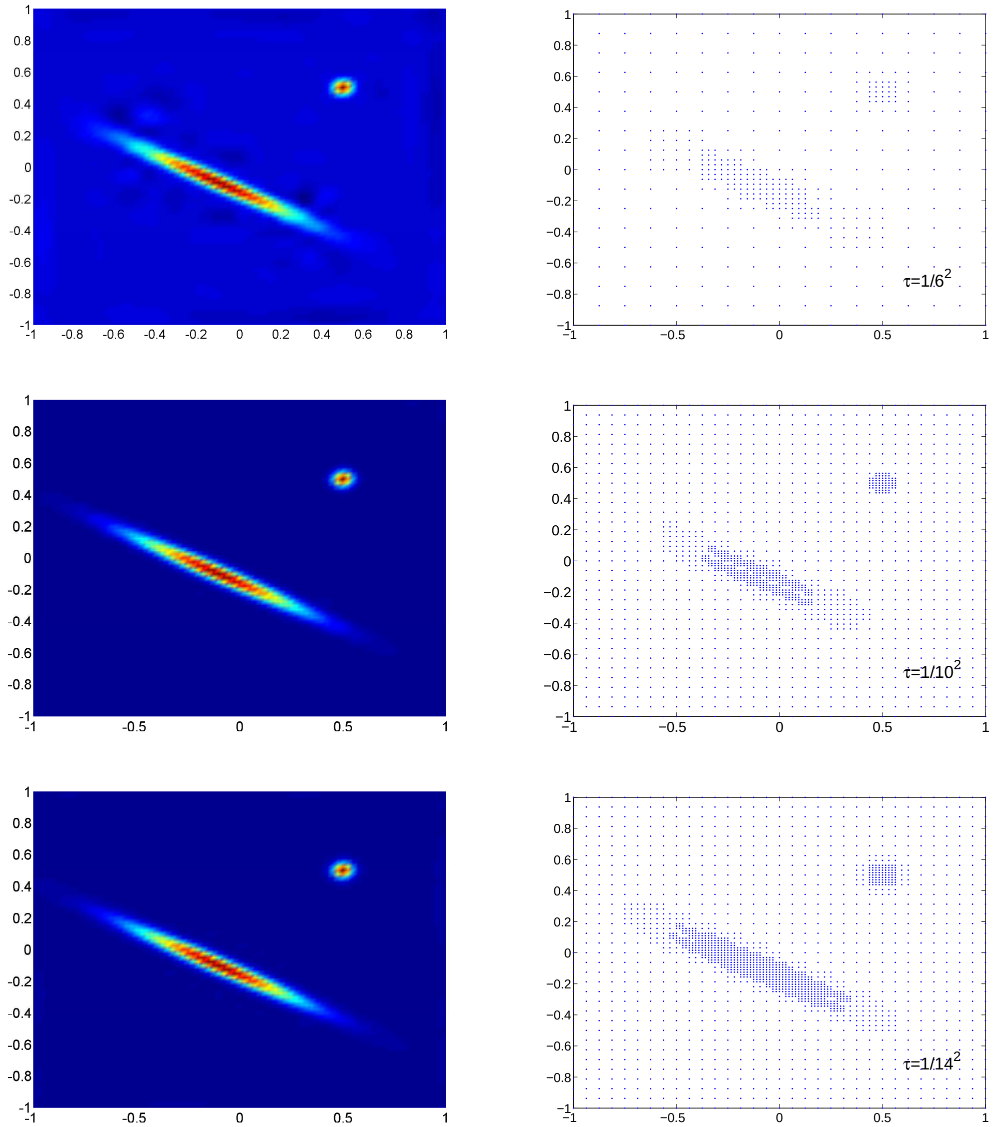

Figure 7. Example 4.3: Hybrid approximations and the corresponding adaptive nodes distributions (before *greedy* selections) for various principle tolerances τ.

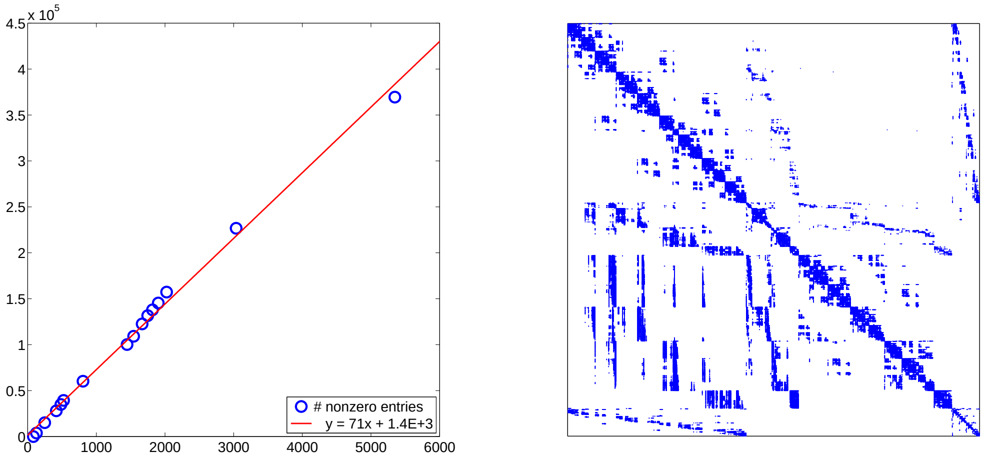

Figure 8. Example 4.3: (Left) Number of nonzero entries in State II against the numbers of data sites; (Right) sparsity pattern of the collocation matrix in Stage II with $\tau = 1/14^2$.

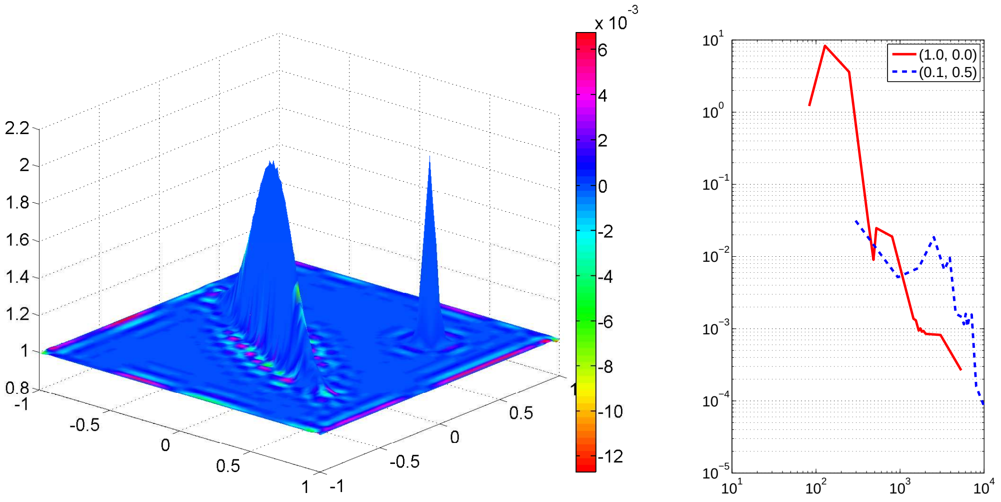

Figure 9. Example 4.3: (Left) a numerical approximation to the elongated near-singularities; (Right) relative $L^2(\Omega)$ errors $\mathcal{E}_2$ with different (c, α) against the number of data sites N

.

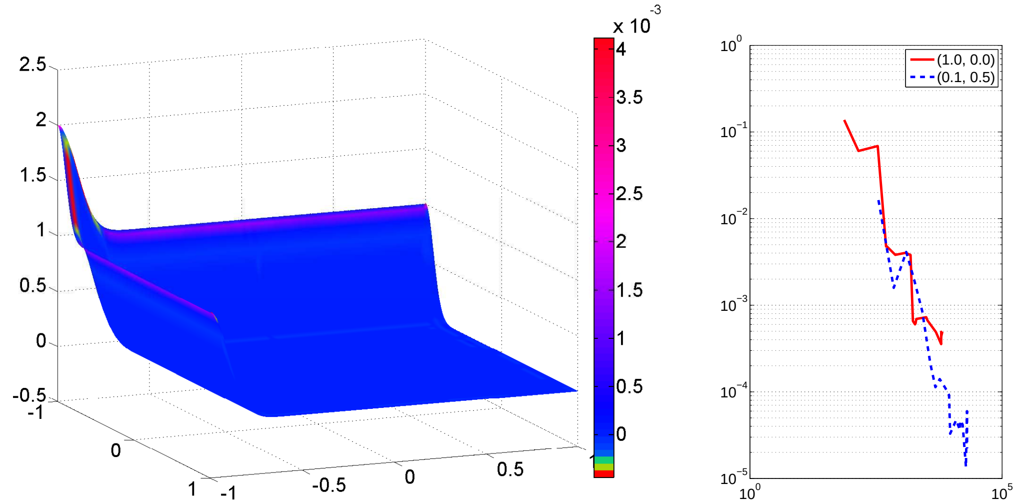

Figure 10. Example 4.4: (Left) a numerical approximation to the boundary layers; (Right) relative $L^2(\Omega)$ errors $\mathcal{E}_2$ with different (c, α) against the number of data sites N.

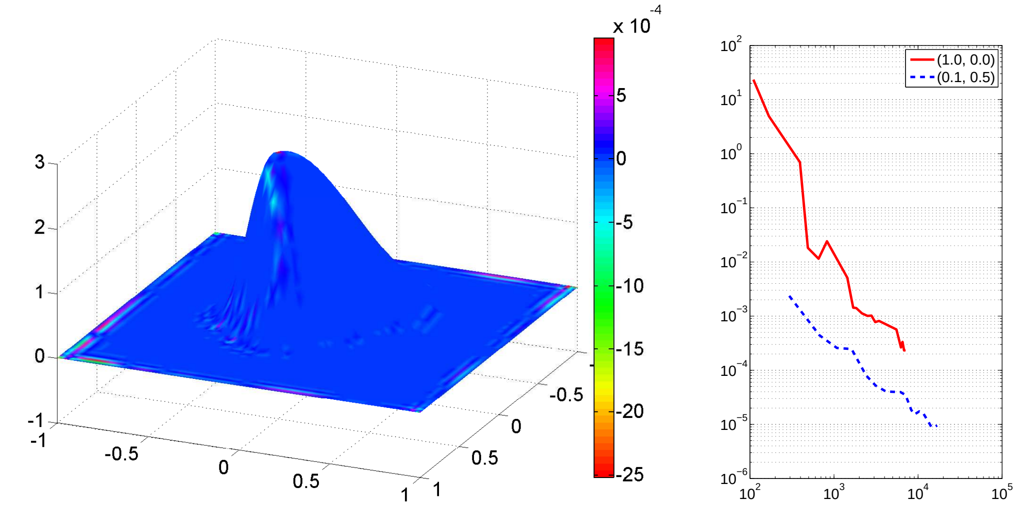

Figure 11. Example 4.5: (Left) a numerical approximation to the shock formation; (Right) relative $L^2(\Omega)$ errors $\mathcal{E}_2$ with different (c, α) against the number of data sites N.

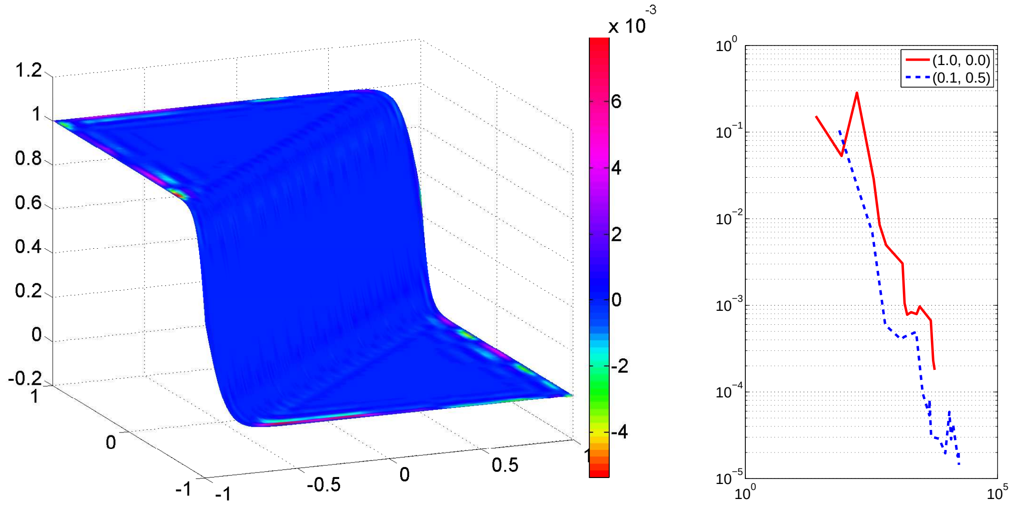

Figure 12. Example 4.6: (Left) a numerical approximation to moving front; (Right) relative $L^2(\Omega)$ errors $\mathcal{E}_2$ with different (c, α) against the number of data sites N.

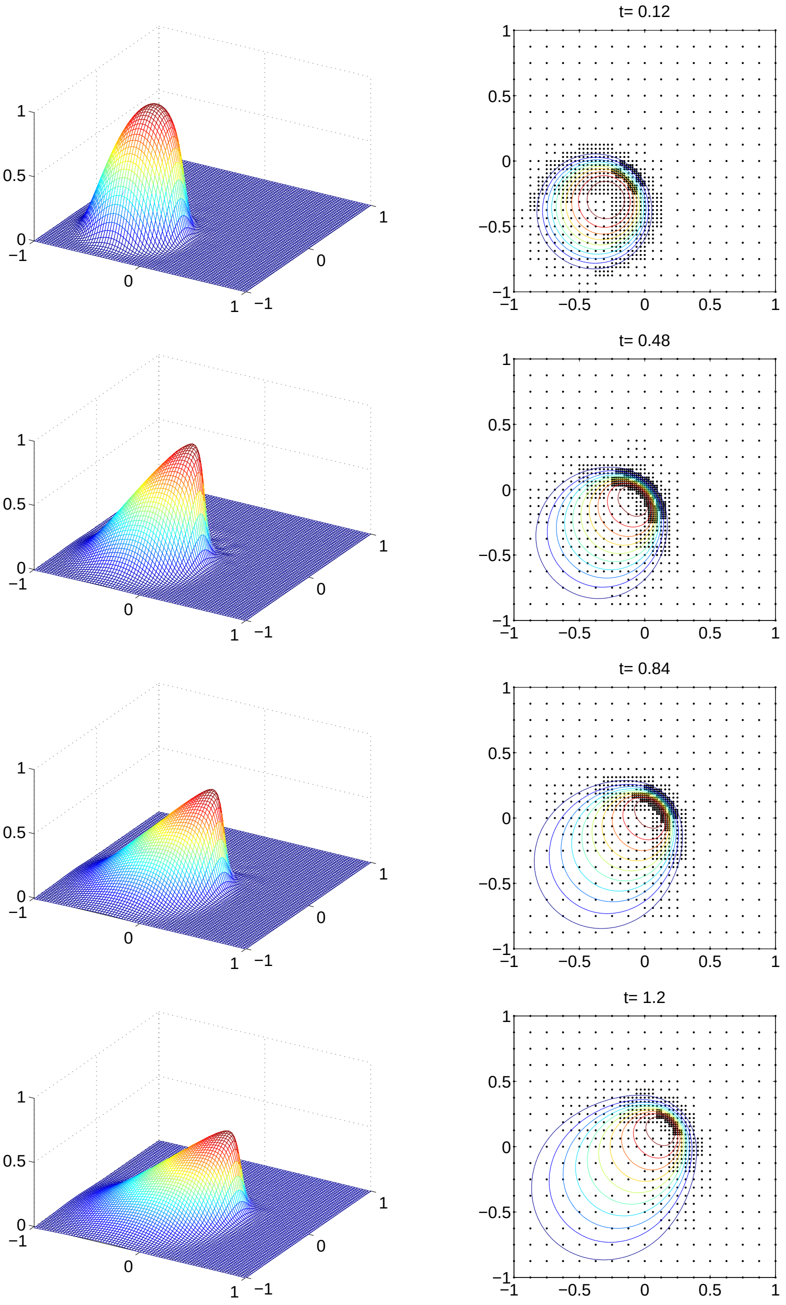

Figure 13. Example 4.7: (Left) Numerical solutions to the Burgers' equation; (Right) Adaptively placed data sites and solution contours for various time t.