Results on Meshless Collocation Techniques

Leevan Ling [1], Roland Opfer [2] and Robert Schaback [2]

Abstract

Though the technique introduced by Kansa is very successful in engineering applications, there were no proven results so far on the unsymmetric meshless collocation method for solving PDE boundary value problems in strong form. While the original method cannot be proven to be fail–safe in general, we prove asymptotic feasibility for a generalized variant using separated trial and test spaces. Furthermore, a greedy variation of this technique is provided, allowing a fully adaptive matrix–free and data–dependent meshless selection of the test and trial spaces.

Key words: Kansa's Methods, Asymptotic Feasibility, Matrix-free Algorithm

1 Introduction

The general idea for solving PDE problems in strong or weak form by kernel–based meshless methods was outlined in [15]. It writes the PDE problem as an uncountably infinite number of simultaneous scalar equations

$$\lambda[u] = f_\lambda \in \mathbb{R}, \text{ for all } \lambda \in \Lambda. \tag{1}$$

[1] Department of Applied Mathematics, National Sun Yat-sen University, 70 Lienhai Rd. Kaohsiung 804, Taiwan ROC.

[2] Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, Lotzestraße 16-18, D-37083 Göttingen, Germany.

9 August 2005

The set Λ consists of infinitely many linear real–valued functionals λ that usually take the form of point evaluations of functions or derivatives at points inside a domain or on some boundary or interface layer. If several differential or boundary operators are involved, we simply put everything into a single set Λ of functionals of various types.

We call (1) a *generalized interpolation problem*, and we assume it to be solvable by a function u^* that generates the data via $f_\lambda := \lambda(u^*)$ for all $\lambda \in \Lambda$. Discretization just consists in replacing the infinite set Λ by some finite unstructured subset $\Lambda_n := \{\lambda_1, \ldots, \lambda_n\}$. The space spanned by these functionals can be called the *test space*, and Λ is the infinite *test set*.

The *trial space* consists of a space of functions in which the numerical solution u is constructed, and we assume it to be spanned by a basis $\{u_1, \ldots, u_n\}$ as

$$u := \sum_{j=1}^{n} \alpha_j \, u_j \ \in U := \mathrm{span}\{u_1, \ldots, u_n\}. \tag{2}$$

Then, the discretized problem reads as

$$\lambda_i[u] = \sum_{j=1}^{n} \alpha_j \, \lambda_i[u_j] = f_{\lambda_i} = \lambda_i(u^*), \ 1 \le i \le n, \tag{3}$$

when written as linear equations for a function u of the trial space U.

Weak formulations use functionals of the form $\lambda_j[g] := a_j(g, u_j)$ with certain bilinear forms a_j and *test functions* u_j such that the discretized problem takes the familiar form

$$\lambda_i[u] = a_i(u, u_i) = \sum_{j=1}^{n} \alpha_j \, a_i(u_j, u_i) = f_{\lambda_i} = \lambda_i(u^*), \ 1 \le i \le n,$$

of meshless Petrov–Galerkin schemes [2]. In particular, weak formulations always have a strong built–in connection of test functionals to test functions.

For problems in strong formulation, the connection between test functionals and test functions is to be established differently. To get a truly meshless technique, and to allow very general problems, we use a symmetric positive definite kernel $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

The special case of *symmetric collocation* now takes the discretized set Λ_n of test functionals and defines the trial functions as $u_j := \lambda_j^y \Phi(\cdot, y)$ for $1 \le j \le n$ where the superscript of λ indicates the variable of Φ on which the functional operates. Then the collocation matrix (3) is symmetric with entries $\lambda_i[u_j] = \lambda_i^x \lambda_j^y \Phi(x, y)$ for $1 \le i, j \le n$. This technique dates back to [17] and has a solid mathematical basis [5,6]. Like in the standard (non–Petrov) Galerkin scheme [3], the trial and test functions or functionals are closely related.

If one takes a set $X_n := \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ of scattered points, one can use the trial space U spanned by the trial functions

$$u_j = \Phi(x_j, \cdot), \ 1 \le j \le n, \tag{4}$$

associated to simple point evaluation functionals λ_{δ_j}. Usually, these centers are irregularly placed within Ω. Since the scattered points determine the trial functions, we can call them *trial centers*. This leads to the unsymmetric collocation technique started by Kansa [9,10] for the multiquadric kernel and used by many authors afterwards; for instance, see [4,11–13,16,18] and an overview in [7]. The resulting unsymmetric collocation matrix has the entries

$$\lambda_i[u_j] = \lambda_i^y \Phi(x_j, y), \ 1 \le i, j \le n,$$

and can be singular in exceptional cases [8]. Consequently, there are no mathematical results on this technique, though it gives very good results in plenty of applications in science and engineering.

To overcome these problems partially, one has to modify the setting. To get solvability and error bounds, there should at least be a unique solution to the modified discretized system that converges to the true solution if the discretization is refined. The first question requires that if the n test functionals are fixed and are linearly independent, the system will have rank n provided the trial functions are chosen properly. We shall prove this fact in Theorem 1 below. Then we show how to find proper trial functions in practice via QR factorization once we fixed the test space. Finally, we propose an adaptive greedy method that generates a sequence of nonsingular problems through a search of proper test functionals and trial functions.

2 Nonsingularity

Let $\Omega \subseteq \mathbb{R}^d$ be a domain and $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a *symmetric positive definite kernel* on $\mathbb{R}^d$. The kernel has an associated *native* [14] Hilbert space $\mathcal{N}_\Phi$ of functions on Ω in which it acts as a reproducing kernel [1]. Let a problem of the type (1) on a bounded domain $\Omega \subset \mathbb{R}^d$ be discretized by n linearly independent functionals $\Lambda_n := \{\lambda_1, \ldots, \lambda_n\}$. A suitable kernel for a given PDE must be smooth enough to guarantee $\Lambda \subset \mathcal{N}_\Phi{}^*$. Then the functions $u_\lambda := \lambda^y \Phi(\cdot, y)$ are smooth enough to be in $\mathcal{N}_\Phi$ for all $\lambda \in \Lambda_n$. This is true for plenty of applications, because the smoothness requirements are weaker than those discussed in [6] for symmetric collocation.

Note that *conditionally positive definite* kernels like the multiquadrics can be modified [14] to turn into positive definite kernels by subtracting certain low–order polynomials. This modification allows to restrict theoretical questions to the positive definite case, but in practical implementations one will often

prefer to carry the additional polynomials along. This requires some additional linear algebra that we omit here.

The given discretized strong collocation problem of (1) consists in finding a function in U such that the equations (3) are satisfied for a set Λ_n of linearly independent continuous linear functionals on U and prescribed real values $f_{\Lambda_n} := (f_{\lambda_1}, \ldots, f_{\lambda_n})^T$. Usually, the functionals are of different types. However, we keep the situation as general as possible, allowing quite arbitrary functionals.

The standard trial space for Kansa's unsymmetric collocation method is the span of functions $\Phi(x_j, \cdot)$ for a set $X_n := \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ of suitably placed *trial centers*. The system (3) then has the (in general unsymmetric) $n \times n$ matrix

$$A_{\Lambda_n, X_n} := \left[\lambda_i^y \, \Phi(x_j, y) \right]_{1 \le i,j \le n} \tag{5}$$

for $\lambda_i \in \Lambda_n$ and $x_j \in X_n$. One then solves

$$A_{\Lambda_n, X_n} \, \alpha = f_{\Lambda_n}, \tag{6}$$

for the unknown coefficients $\alpha := \{\alpha_1, \ldots, \alpha_n\}$. Numerical evidence shows that cases of singularity are extremely rare, suggesting that A_{Λ_n, X_n} is nonsingular for most choices of X_n.

Now, we consider the problem from a different perspective: suppose the set of test functionals Λ_n is given, can one find a set of trial centers X_n such that the Kansa's collocation matrix (5) is nonsingular? We begin with a general result:

Lemma 1 *Let $\Omega \subseteq \mathbb{R}^d$ be a domain and let $\{g_1, \ldots, g_n\}$ for $1 \le i \le n$ be a linearly independent set of n continuous functions defined on Ω. For a*

sufficiently dense subset $X = \{x_1, \ldots, x_M\}$ *of points in* Ω*, the matrix with entries* $g_i(x_j)$ *for* $1 \le i \le n$*, and* $1 \le j \le M$ *has full rank* n*.*

Proof: Take an increasing sequence $\{X^M\}_M$ of data sets $X^M := \{x_1, \ldots, x_M\} \subset \Omega$ such that the union of these sets is dense in Ω. If the assertion is false, there are nonzero vectors $\alpha^M := (\alpha_1^M, \ldots, \alpha_n^M) \in \mathbb{R}^n$ such that

$$\sum_{i=1}^{n} \alpha_i^M \, g_i(x_j) = 0, \; 1 \le j \le M,$$

holds for all M. Since we can renormalize each vector in the set $\{\alpha^M\}_M$ to have norm one, we can find a subsequence that converges to a nonzero normalized vector $\tilde{\alpha} \in \mathbb{R}^n$. If we define the functions

$$s_\alpha := \sum_{i=1}^{n} \alpha_i \, g_i,$$

for all $\alpha \in \mathbb{R}^n$, we get

$$s_{\tilde{\alpha}}(x_i) = \Big(s_{\tilde{\alpha}}(x_i) - s_{\alpha^M}(x_i) \Big) + s_{\alpha^M}(x_i),$$

and the right–hand side tends to zero for $M \to \infty$ and all fixed points x_i. Consequently, the function $s_{\tilde{\alpha}}$ vanishes on a dense subset of Ω, and on all of Ω by continuity. But its L_2-norm defined via the Gram matrix

$$\|s_\alpha\|_2 := \sum_{i,j=1}^{n} \alpha_i \, \alpha_i \, G_{ij}, \text{ where } G_{ij} := \int_\Omega g_i \, g_j \, dx,$$

is positive since $\tilde{\alpha}$ is nonzero and the $\{g_i\}$ are linearly independent, yielding a contradiction. $\diamond$

Lemma 2 *Under the same hypothesis, there must exists* n *points* $\{x_1, \ldots, x_n\} \in \Omega$ *such that the matrix with entries* $g_i(x_j)$ *for* $1 \le i, j \le n$ *is nonsingular.*

Proof: It is a contradiction to Lemma 1 if such a set does not exist. $\diamond$

To make use of Lemma 2, we construct a set of functions using the test functionals Λ_n and the symmetric positive definite kernel Φ. Let $g_i : \mathbb{R}^d \to \mathbb{R}$ be the associated continuous functions defined as

$$g_i := \lambda_i^y \, \Phi(\cdot, y), \ 1 \le i \le n.$$

Then, the Kansa's collocation matrix can be expressed as

$$A_{\Lambda_n, X_n} = [\, g_i(x_j) \,]_{1 \le i,j \le n}.$$

Theorem 1 *Assume the kernel Φ to be smooth enough to guarantee that the functions $u_\lambda := \lambda^y \, \Phi(\cdot, y)$ for $\lambda \in \Lambda$ are continuous. Then the set of functions $\{g_i\}_{i=1}^n$ constructed above is linearly independent, and hence Kansa's collocation matrix* (5) *is nonsingular for properly chosen trial centers.*

Proof: By assumption, all $\{g_i\}$ are continuous. After we prove the independence, the second assertion follows from Theorem 2. If $\sum_{i=1}^n \beta_i \, g_i = 0$, then by the reproducing property of the kernel

$$0 = \left\langle f, \sum_{i=1}^n \beta_i \, g_i \right\rangle = \sum_{i=1}^n \beta_i \, \lambda_i^y \left\langle f, \Phi(\cdot, y) \right\rangle = \sum_{i=1}^n \beta_i \, \lambda_i[f], \ \text{ for all } \ f \in \mathcal{N}_\Phi,$$

which implies that $\beta_i \equiv 0$, and thus $\{g_i\}$ are linearly independent. $\diamond$

The continuity assumption of Theorem 1 is not satisfied if we handle weak problems by kernels with minimal regularity, e.g. solving a Poisson problem in 2-dimension using the kernel K_0 that reproduces the Sobolev space $W_2^1(\mathbb{R}^2)$.

In practice, one should fix the set Λ_n of n test functionals first and then work on a trial space U constructed via a very large set X_M of trial centers located in a bounded domain that contains Ω. If $M \gg n$ is large enough, and if the trial centers are reasonably distributed, the set X_M contains a proper set of trial centers as guaranteed by Theorem 1. The employed method should

automatically pick n out of these M centers to guarantee nonsingularity and, most importantly, a reasonable condition of the resulting $n \times n$ collocation matrix. In terms of linear algebra, this means to pick a subset X_n of n points out of X_M such that the square $n \times n$ submatrix A_{Λ_n, X_n} of the large non–square $n \times M$ matrix A_{Λ_n, X_M} is nonsingular. Clearly, this can be done by a partial QR factorization of A_{Λ_n, X_M} with column pivoting. The computational complexity grows as $\mathcal{O}(n^2 M)$, because n steps are needed, each updating an $n \times M$ matrix by an $n \times n$ Householder transformation. Storage grows like $\mathcal{O}(nM)$. But there are techniques that gradually build up the relevant matrix "on–the–fly", as we shall show below.

3 Greedy Method

The previous section showed how to deal with a fixed set of test functionals by picking suitable subsets of trial centers in a meshless and data–dependent way. But choosing a set Λ_n that best approximates the original infinite test set Λ is also essential. In view of the infinite problem (1) one should also pick suitable *test functionals* in a meshless and data–dependent way from a large set of test functionals. The choice of trial centers is left to a later stage. In this section, we provide an adaptive greedy algorithm that automatically searches for a suitable test/trial pair at each iteration consisting of a test functional λ_{k+1} and a trial center x_{k+1}. It will work on $k \times k$ systems, allowing $k < n$ to grow, and doing update steps of complexity $\mathcal{O}(k^2)$ after having found the pair (λ_{k+1}, x_{k+1}). This even allows to work on infinite sets of test functionals and trial centers to choose from. We call the above property *matrix-free* since the evaluation of the full (finite/infinite) matrix is unnecessary.

Given an infinite set $\Lambda \subseteq {\mathcal{N}_\Phi}^*$ of functionals. We want to reconstruct a function $u \in \mathcal{N}_\Phi$ from its data Λ. At iteration k, assume that a solution to Kansa's method for functionals $\Lambda_k := \{\lambda_1, \ldots, \lambda_k\}$ and suitably placed trial centers $X_k := \{x_1, \ldots, x_k\} \subset \mathbb{R}^d$ is already known such that the corresponding matrix is nonsingular. We write this as the $k \times k$ matrix system

$$A_{\Lambda_k, X_k}\, \alpha^k = f_{\Lambda_k}, \tag{7}$$

and denote the solution function by $s_k := \sum_{i=1}^{k} \alpha_i^k\, \Phi(\cdot, x_i)$.

Now we pick from the infinite set Λ a functional λ_{k+1} such that the residual $\lambda_{k+1}(s_k - u)$ is large in absolute value, possibly maximal among all other such functionals. The technique is therefore called *greedy*.

It suffices to assume

$$\Big|\lambda_{k+1}(s_k - u)\Big| \geq \theta \Big|\lambda(s_k - u)\Big| \text{ for all } \lambda \in \Lambda, \tag{8}$$

for some $\theta \in (0, 1]$, e.g., by taking the actual maximum. If we find none with a nonzero value, we stop. Otherwise we conclude that λ_{k+1} must be linearly independent from the other functionals. Now add λ_{k+1} to the functionals considered so far,

$$\Lambda_{k+1} := \Lambda_k \cup \{\lambda_{k+1}\}.$$

For a moment, we consider the new trial center x_{n+1} as a free variable $x \in \mathbb{R}^d$. The determinant of $A_{\Lambda_{k+1}, X_{k+1}}$ is then a function $v_{k+1}(x)$, in fact up to a sign

$$\begin{aligned} |v_{k+1}(x)| &= \Big|\det\Big(A_{\Lambda_{k+1}, X_k \bigcup \{x\}}\Big)\Big| \\ &= \left|\sum_{i=1}^{k+1} (-1)^i\, g_i(x)\, \det\Big(A_{\Lambda_{k+1}\setminus\{\lambda_i\}, X_k}\Big)\right|, \end{aligned}$$

where $g_i := \lambda_i^y \, \Phi(\cdot, y)$. If we define $\beta_i = (-1)^i \det\Big(A_{\Lambda_{k+1}\setminus\{\lambda_i\},X_k}\Big)$, we have

$$|v_{k+1}(x)| = \left| \sum_{i=1}^{k} \beta_i \, g_i(x) + (-1)^{k+1} \det\Big(A_{\Lambda_k,X_k}\Big) g_{k+1}(x) \right|. \tag{9}$$

Now we pick a point $x^* \in \mathbb{R}^d$ such that $v_{k+1}(x^*) \neq 0$ and $|v_{k+1}(x^*)|$ is closest to 1; call it x_{k+1} and add it to the previously chosen set of trial centers,

$$X_{k+1} := X_k \cup \{x_{k+1}\}.$$

Now repeat the iteration for $k+1$ instead of k. In the step that selects the functional λ_{k+1} we have something similar to the well-known a–posteriori error estimators in finite elements. But, in addition, our subsequent choice of the trial center x_{k+1} *adaptively* changes the trial space in a data–dependent and meshless way.

Theorem 2 *The adaptive greedy algorithm above generates a sequence of uniquely solvable unsymmetric collocation problems. It can be carried out efficiently, since the factors β_i in* (9) *are constant multiples of the solution of*

$$A_{\Lambda_k,X_k}^T \, \gamma = \Big(g_{k+1}(x_1), \ldots, g_{k+1}(x_k) \Big)^T. \tag{10}$$

Proof: For the functional $\lambda_1 \in \Lambda_n$, it is clear that there must exist a point $x_1 \in \mathbb{R}^d$ such that $g_1(x_1) \neq 0$. Suppose $\det\Big(A_{\Lambda_k,X_k}\Big) \neq 0$. We assume that $v_{k+1}(x) = 0$ for all $x \in \mathbb{R}^d$. Since the functions g_i for $1 \leq i \leq n$ are linearly independent, all β_i for $1 \leq i \leq k+1$ in (9), including $\beta_{k+1} := \det\Big(A_{\Lambda_k,X_k}\Big)$, are identically zero. This yields a contradiction to Lemma 1.

We apply Cramer's rule to the system (10) and get

$$\gamma_i = (-1)^{k-i} \, \frac{\det\Big(A^T_{\Lambda_{k+1}\setminus\{\lambda_i\},X_k}\Big)}{\det A_{\Lambda_k,X_k}}, \quad 1 \leq i \leq k,$$

for the solution, and formally also $\gamma_{k+1} = -1$. Thus, up to the irrelevant nonzero common factor $\sigma_k := \pm \det A_{\Lambda_k, X_k}$, we have $\beta_i = \sigma_k \, \gamma_i$, for $1 \le i \le k+1$. $\diamond$

In practice, we use the scaled function $v_{k+1}(x)/\sigma_{k+1} = \sum_{i=1}^{k+1} \gamma_i \, g_i(x)$ instead.

4 Implementation and Cost

The adaptive greedy method, if applied to a huge but finite linear system, can be reformulated in linear algebra terms. Then, it is different from standard elimination techniques with pivoting, because it takes the right–hand side into account. It also has an "on–the–fly" formulation which we describe now.

Within each iteration, instead of solving (7) and (10) from scratch, we can update the inverse matrices through a well–known matrix inversion formula. Let B_k denote the inverse of the k-th stage matrix A_{Λ_k, X_k}. For $k > 1$, we have

$$B_k = \begin{bmatrix} A_{\Lambda_{k-1}, X_{k-1}} & \vec{v} \\ \vec{u}^T & w \end{bmatrix}^{-1} \qquad (11)$$

$$= \begin{bmatrix} E + \delta \, (B_{k-1} \vec{v})(\vec{u}^T \, B_{k-1}) & -\delta \, (B_{k-1} \, \vec{v}) \\ -\delta \, (\vec{u}^T \, B_{k-1}) & \delta \end{bmatrix},$$

where the vectors and scalars above are given by $\vec{u} = g_k(x_j)$, $\vec{v} = g_i(x_k)$ for $1 \le i, j \le k-1$, $w = g_k(x_k)$, $\delta := (w - \vec{u}^T \, B_{k-1} \, \vec{v})^{-1}$, and E denotes the identity matrix of size $k-1$. The inverse update algorithm requires at most $\mathcal{O}(k^2)$ operations, and it can be viewed as the recursive form of Gaussian

elimination.

In practice, we discretize the generalized PDE problem first via a large set of N test functionals Λ_N. Then we provide a very large set of $M \gg N$ trial centers X_M for the algorithm to choose from. Note that the test functionals are restricted to a computational domain and its boundaries, while the trial centers can come from a larger domain.

The algorithm can be viewed as working on a full $N \times M$ matrix, but it does never compute or store the full matrix. Instead, only the essential elements are computed and stored "on–the–fly".

Algorithm:

Startup is done by finding some $\lambda_1 \in \Lambda_N$ with the greedy criterion (8) and then an $x_1 \in X_M$ such that $\mathbb{R} \ni g_1(x_1) := \lambda_1^y \Phi(x_1, y) \neq 0$ and large in absolute value. The first inverse matrix and the first approximated solution are given by

$$B_1 := g_1(x_1)^{-1} \in \mathbb{R}, \; s_1 = \frac{f_{\lambda_1}}{g_1(x_1)} g_1(\cdot).$$

For some user defined integers $1 < k_{\text{restart}} < k_{\max}$, the algorithm then iterates on $k = 2, 3, \ldots$ as follows:

(1) Compute all residuals $\lambda_i(s_{k-1} - u)$ for all $\lambda_i \in \Lambda_N$; this takes $\mathcal{O}(kN)$ operations.

(2) Search for $\lambda_k \in \Lambda_N$ with maximum residual; this takes $\mathcal{O}(N)$ operations.

- STOP if all residuals in absolute value are smaller than some tolerance.

(3) Associated to λ_k is a function $g_k = \lambda_k^y \Phi(\cdot, y)$. Compute and store $g_k(x_j)$ for $1 \leq j \leq M$. This takes $\mathcal{O}(M)$ operations.

(4) Evaluate the function

$$\frac{v_k(x)}{\sigma_k} = -g_k(x) + \Big[g_k(x_1), \ldots, g_k(x_{k-1})\Big] B_{k-1} \begin{bmatrix} g_1(x) \\ \vdots \\ g_{k-1}(x) \end{bmatrix},$$

for all $x \in X_M$ (see Theorem 2). This takes $\mathcal{O}(kM)$ operations.

(5) Search for $x_k \in X_M$ such that $|v_k(x_k)/\sigma_k|$ is closest to 1. This takes $\mathcal{O}(M)$ operations.

- STOP if computed values are smaller than some tolerance.

(6) Update the index sets corresponding to the new sets of test functionals and trial centers, $\Lambda_k = \Lambda_{k-1} \bigcup \{\lambda_k\}$ and $X_k = X_{k-1} \bigcup \{x_k\}$, respectively.

(7) IF $\mathrm{mod}(k, k_{\mathrm{restart}}) = 0$, OR $k = k_{\max}$, solve subsystem (7),

$$A_{\Lambda_k, X_k}\, \alpha^k = f_{\Lambda_k},$$

with utmost numerical care, using SVD. Then STOP.

(8) ELSE update the inverse matrix B_k as in (11). This takes $\mathcal{O}(k^2)$ operations.

(9) Solve the $k \times k$ subsystem (7) for $\alpha^k = B_k\, f_{\Lambda_k}$. This takes $\mathcal{O}(k^2)$ operations.

(10) Repeat the iteration for $k + 1$.

The rule in step 5 may look strange, but since the function v_k is (up to a factor) the determinant of the enlarged system, one has to avoid large *and* small values to ensure numerical stability.

If the final iteration count is K, the total work of the adaptive greedy method is $\mathcal{O}(K^3 + K^2 M + K^2 N)$ operations, while storage is of order $\mathcal{O}(K^2 + M + N)$.

The method is extremely efficient in theory and practice, if K is small and M, N are large. The final size of K determines its overall complexity, and K will be strongly problem–dependent, turning out to be surprisingly small in standard applications, as we will show in the next section.

5 Numerical results

In this section we show some numerical examples which demonstrate the efficiency of our proposed adaptive greedy algorithm. In all presented examples we have used the multiquadric

$$\Phi_c(x,y) = \sqrt{1 + \frac{||x-y||^2}{c^2}},$$

where $x, y \in \mathbb{R}^2$ and $c > 0$ is the *scaling parameter*. As test equation we solve the Poisson problem with Dirichlet boundary conditions, i.e.

$$\begin{aligned} \triangle u(x) &= f(x) \text{ for } x \in \Omega \subset \mathbb{R}^d, \\ u(x) &= g(x) \text{ for } x \in \partial\Omega. \end{aligned} \tag{12}$$

We solve this problem on the three domains Ω_1, Ω_2, $\Omega_3 \subset \mathbb{R}^2$ which are plotted in figure 1.

In order to compare the exact solution of (12) with the computed approximant, we choose as the right hand sides of (12) the functions $f = \triangle g_i$ for $1 \leq i \leq 3$ where the functions g_i are

$$\begin{aligned} g_1(x,y) &:= 0.5 \log(x^2 - 4x + 8 + y^2 - 4y), \\ g_2(x,y) &:= \text{ MATLAB's peaks function}, \\ g_3(x,y) &:= (\max\{x, 0\})^3. \end{aligned}$$

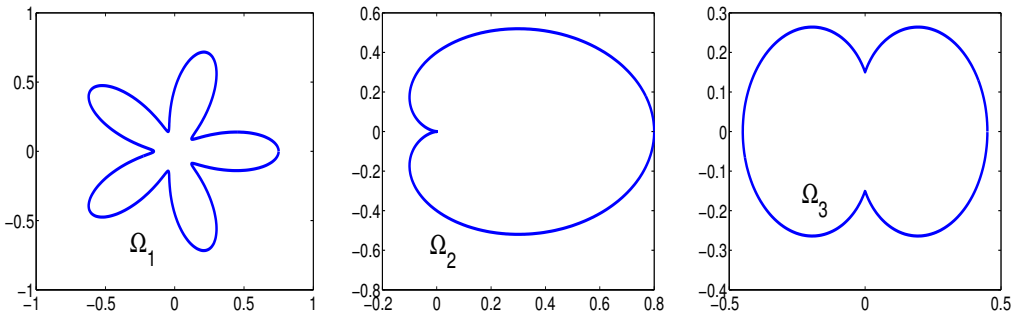

Fig. 1. The domains Ω_1, Ω_2, and Ω_3.

We collocate the PDE (12) at $N_1 + N_2$ points

$$\{z_1, \ldots, z_{N_1}\} \subset \overline{\Omega}, \quad \{\tilde{z}_1, \ldots, \tilde{z}_{N_2}\} \subset \partial\Omega \tag{13}$$

such that the first point set associates with the domain operator and the second set associates with the boundary operator.

Fill distance of the trial points should be more or less equal to that of the test points. The choice of the outer box, in which the trial points lies, determines the value of M and vise versa. We discretize the set $[-6, 6]^2$ by 47229 equally distributed points $X_M \subset [-6, 6]^2$. We allow the greedy method to pick from the trial functions

$$\Big\{\Phi_c(x_i, \cdot) : x_i \in X_M\Big\},$$

and from the test functionals

$$\Big\{\lambda_i := \delta_{z_i} \triangle[u] : 1 \leq i \leq N_1\Big\} \cup \Big\{\tilde{\lambda}_j := \delta_{\tilde{z}_j}[u] : 1 \leq j \leq N_2\Big\}.$$

Finally, we pick $k_{\text{restart}} = 10$ and $k_{\max} = 250$.

Since the method is extremely fast, the forthcoming examples are not restricted to single solutions of single problems. Instead, a full sequence of problems is solved, using varying scalings of the kernel.

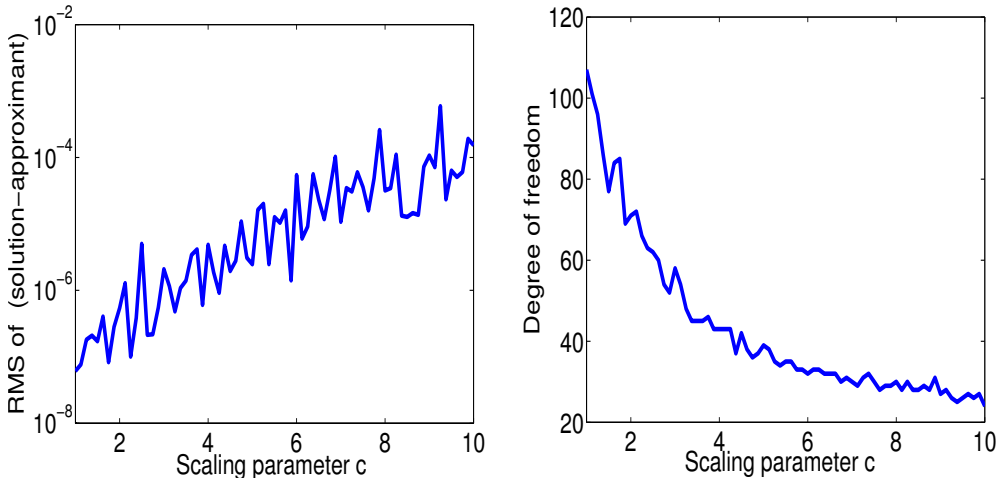

Fig. 2. RMS and DOF against scaling parameter c for the domain Ω_1 and for $f = \triangle g_1$.

Example 1 In our first example we solve (12) for the domain Ω_1 (see Figure 1) and for $f = \triangle g_1$. The adaptive greedy algorithm can pick from $N_1 + N_2 = 1116$ test functionals (see (13)). To show the influence of the scaling parameter c, we run the algorithm 73 times with $c_i := 1 + 0.125i, 0 \leq i \leq 72$. The left plot of figure 2 shows the root-mean-square (RMS) error against the scaling parameter c;

$$\text{RMS} := \frac{1}{|\tilde{\Omega}|}\sqrt{\sum_{x\in\tilde{\Omega}}(s_1(x) - \tilde{s}(x))^2}$$

where $\tilde{s}$ denotes the computed approximant and $\tilde{\Omega}$ is a sufficiently fine discretization of Ω The right plot of figure 2 shows the degrees of freedom, i.e. the number K of chosen trial centers or the number of total iterations, against the scaling parameter c.

A small value of c allows the algorithm to pick many trial centers and test functionals which results in a high accuracy. A huge value of c causes an ill–conditioned matrix. Therefore the iteration terminates after fewer steps, i.e.

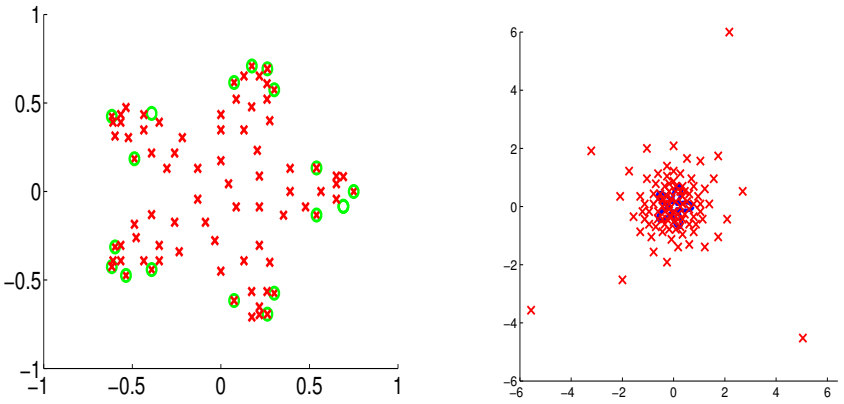

Fig. 3. Loci of points for $c = 1$. Left: chosen test functionals. Right: chosen trial centers.

better efficiency. However, all computed examples have a remarkably small error, taking into account that we have actually picked less than 120 points in all examples. The greedy algorithm proposed by [8] needs nearly 3000 points for the same problem to achieve an accuracy of 0.001. To show how the scaling parameter c influences the distribution of the points, Figure 3 and 4 show the loci of the chosen trial centers and test functionals for the scaling parameters $c = 1$ and $c = 5$. In the left plots of figure 3 and 4, the circles indicate the chosen boundary functionals whereas the crosses indicate the interior functionals.

For $c = 1$ the trial centers are uniformly distributed close to the domain Ω_1, whereas for $c = 5$ the trial centers (39 out of 40) move out of the domain. We observed the same behaviour in other examples: the adaptive greedy algorithm prefers "exterior" centers for large scaling parameters c.

Example 2 In this example we solve the PDE (12) for the domain Ω_2 (see Figure 1) and for $f = \triangle g_2$. We run the algorithm with the same settings as in the previous example, but the number of test functionals is now $N_1 + N_2 = 1082$. The left plot of figure 5 shows the RMS error while the right plot of

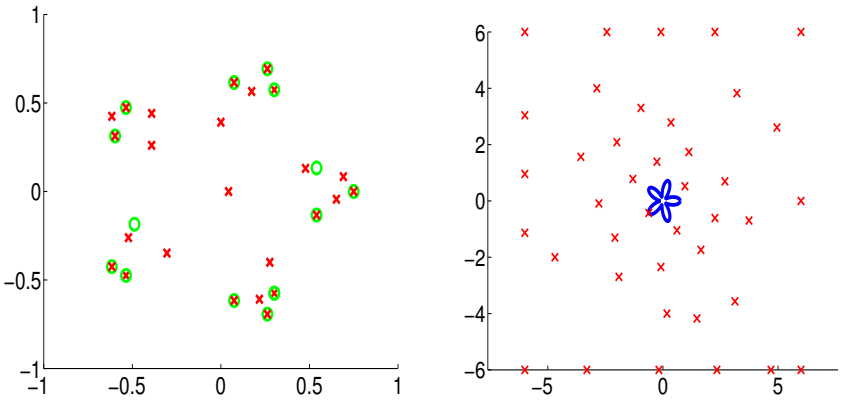

Fig. 4. Loci of points for $c = 5$. Left: chosen test functionals. Right: chosen trial centers.

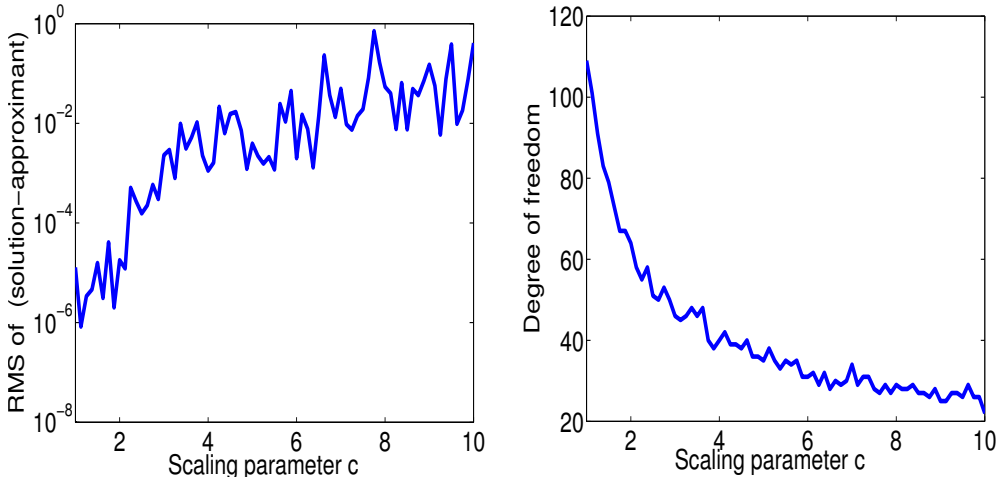

Fig. 5. RMS and DOF against scaling parameter c for the domain Ω_2 and for $f = \triangle g_2$.

figure 5 shows the final degrees of freedom K against the scaling parameter c.

In this example we observe a similar behaviour of the algorithm with respect to the scaling parameter c as in the previous example. A small c results in a very high accuracy. If we choose the scaling parameter close to 10, the algorithm can only pick about 20 points before it runs into condition problems. Therefore

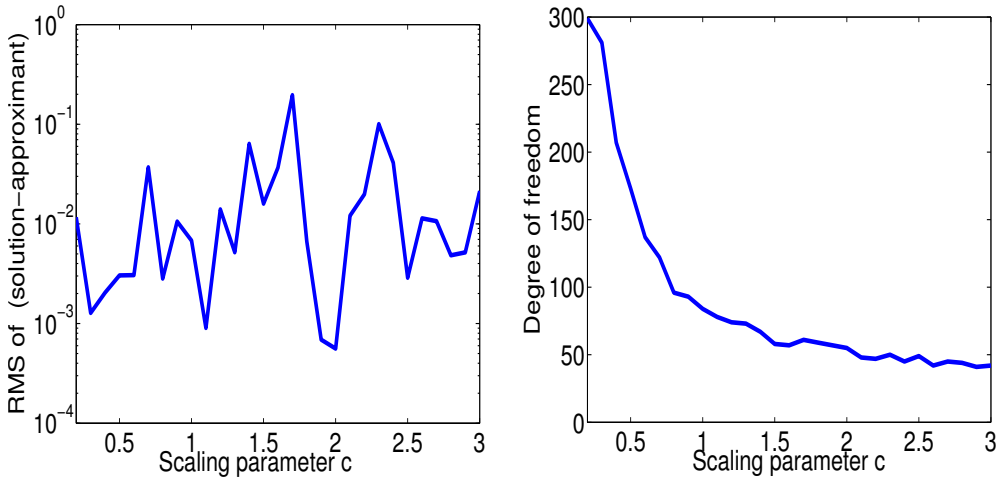

Fig. 6. RMS and DOF against scaling parameter c for the domain Ω_3 and for $f = \triangle g_3$.

we lose accuracy.

Example 3 In our last example we solve the PDE (12) for Ω_3 and for $f = \triangle g_3$. Note that in this example $\triangle g_3$ has a corner inside the domain Ω_3, and the solution has a derivative singularity there. Since our approximant is always a superposition of shifted super smooth functions, we cannot expect a high accuracy. The number of test functionals is $N_1 + N_2 = 636$ in this example. We only ran the adaptive greedy algorithm 29 times for $c_i := 0.2 + 0.1i, 0 \leq i \leq 28$. Figure 6 shows the RMS error and the degrees of freedom against the scaling parameter c.

6 Conclusion

We propose a truly meshless adaptive greedy technique that allows general PDE problems to be solved on complicated domains. The method is adaptive, data-dependent and matrix-free; hence, it provides a fast and efficient

alternative to solve large-scale problems.

Note that the greedy method currently runs on large discrete sets of test functionals and trial centers. But it can be run on infinite sets thereof. In the latter case, the searches over the discrete sets Λ_N and X_M are replaced by two d-dimensional optimization problems. Also, there is quite some chance to prove convergence to the true solution of the full problem, using the techniques of [15]. We leave this to a forthcoming paper, together with the possibility to use trial spaces that arise from kernels of different scales and incorporate functions which account for singularities of solutions at the boundary.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[2] Satya N. Atluri and Shengping Shen. *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Encino, CA, 2002.

[3] F. A. de Paula and J. C. F. Telles. A comparison between point collocation and Galerkin for stiffness matrices obtained by boundary elements. *Eng. Anal. Boundary Elements*, 6(3):123–128, 1989.

[4] G. E. Fasshauer. Newton iteration with multiquadrics for the solution of nonlinear PDEs. *Comput. Math. Appl.*, 43(3-5):423–438, 2002.

[5] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93(1):73–82, 1998.

[6] Carsten Franke and R. Schaback. Convergence order estimates of meshless

collocation methods using radial basis functions. *Adv. Comput. Math.*, 8(4):381–399, 1998.

[7] M.A. Golberg and C.S. Chen, editors. *The dual reciprocity method and radial basis functions*. Elsevier, Amsterdam, 2000. Eng. Anal. Boundary Elements **24** (2000), no. 7-8.

[8] Y. C. Hon and R. Schaback. On unsymmetric collocation by radial basis functions. *Appl. Math. Comput.*, 119(2-3):177–186, 2001.

[9] E. J. Kansa. Application of Hardy's multiquadric interpolation to hydrodynamics. In *Proc. 1986 Simul. Conf., Vol. 4*, pages 111–117, 1986.

[10] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19(8-9):127–145, 1990.

[11] Jichun Li, Alexander H. D. Cheng, and Ching-Shyang Chen. A comparison of efficiency and error convergence of multiquadric collocation method and finite element method. *Eng. Anal. Boundary Elements*, 27(3):251–257, 2003.

[12] Leevan Ling and M. R. Trummer. Multiquadratic collocation method with integral formulation for boundary layer problems. *Comput. Math. Appl.*, 48(5-6):927–941, 2004.

[13] N. Mai-Duy and T. Tran-Cong. Numerical solution of Navier-Stokes equations using multiquadric radial basis function networks. *Int. J. Num. Meth. Fluid*, 37:65–86, 2001.

[14] R. Schaback. Native Hilbert spaces for radial basis functions I. In Buhmann M.D., Mache D. H., Felten M., Müller, and M.W., editors, *New Developments in Approximation Theory*, volume 132 of *International Series of Numerical Mathematics*, pages 255–282. Birkhäuser Verlag, 1999.

[15] R. Schaback. On the versatility of meshless kernel methods. In S.N. Atluri, D.E. Beskos, and D. Polyzos, editors, *Advances in Computational & Experimental Engineering & Sciences*, number 428, 2003.

[16] A. S. M. Wong, Y. C. Hon, T. S. Li, S. L. Chung, and E. J. Kansa. Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme. *Comput. Math. Appl.*, 37(8):23–43, 1999.

[17] Zong Min Wu. Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.*, 8(2):1–10, 1992.

[18] D. L. Young, S. C. Jane, C. Y. Lin, C. L. Chiu, and K. C. Chen. Solutions of 2D and 3D Stokes laws using multiquadrics method. *Eng. Anal. Boundary Elements*, 28(10):1233–1243, 2004.