An Adaptive LOOCV-Based Algorithm for Solving Elliptic PDEs via RBF Collocation

An Adaptive LOOCV-based Algorithm for Solving Elliptic PDEs via RBF Collocation*

R. Cavoretto[0000−0001−6076−4115] and A. De Rossi[0000−0003−1285−3820]

Department of Mathematics "Giuseppe Peano", University of Torino
Via Carlo Alberto 10, 10123 Torino, Italy
{roberto.cavoretto,alessandra.derossi}@unito.it

Abstract. We present a new adaptive scheme for solving elliptic partial differential equations (PDEs) through a radial basis function (RBF) collocation method. Our adaptive algorithm is meshless and it is characterized by the use of an error indicator, which depends on a leave-one-out cross validation (LOOCV) technique. This approach allows us to locate the areas that need to be refined, also including the chance to add or remove adaptively any points. The algorithm turns out to be flexible and effective by means of a good interaction between error indicator and refinement procedure. Numerical experiments point out the performance of our scheme.

Keywords: Meshfree methods · Adaptive algorithms · Refinement techniques · Poisson problems.

1 Introduction

In this paper we present a new adaptive refinement algorithm for solving 2D elliptic partial differential equations (PDEs) such as Poisson type problems. Our adaptive scheme is applied to Kansa's method, which is known as a nonsymmetric radial basis function (RBF) collocation scheme [6]. This approach has spawned several works, which engaged practitioners and scientists coming from many areas of science and engineering. Several adaptive techniques are used for modeling PDE problems through different RBF methods (see e.g. [2,9]). Basically, our adaptive algorithm is characterized by the use of a *leave-one-out cross validation* (LOOCV) technique, which was originally proposed by Rippa [10]. While the LOOCV was introduced to find an optimal value of the RBF shape parameter (see [4]), in this work it is used as an error indicator within the refinement procedure. This strategy enables us to identify which areas need to be

* The authors acknowledge support from the Department of Mathematics "Giuseppe Peano" of the University of Torino via Project 2019 "Mathematics for applications". Moreover, this work was partially supported by INdAM – GNCS Project 2019 "Kernel-based approximation, multiresolution and subdivision methods and related applications". This research has been accomplished within RITA (Research ITalian network on Approximation).

refined by adding new points. After that phase, we further refine the discretization points based on information arising from another error estimate. This stage of adaptive refinement extends the residual subsampling method proposed in [3], thus allowing us to get an improvements in terms of accuracy. In our numerical experiments we show the performance of our algorithm, which is tested by solving a few Poisson problems. For solving large scale problems one could also think of applying our LOOCV-based scheme to local collocation methods [12].

The paper is organized as follows. In Section 2 we briefly describe Kansa's RBF collocation method. In Section 3 we present the adaptive refinement scheme proposed for solving elliptic PDE problems. Section 4 illustrates numerical experiments carried out to show the performance of our algorithm.

2 Kansa's Collocation Method

Given a domain $\Omega \subset \mathbb{R}^d$ and a linear elliptic partial differential operator $\mathcal{L}$, we define a PDE of the form

$$\mathcal{L}u(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \tag{1}$$

with Dirichlet boundary conditions

$$u(\boldsymbol{x}) = g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial\Omega. \tag{2}$$

For Kansa's approach we express the approximate solution $\hat{u}$ as a linear combination of basis functions as commonly happens for RBF interpolation (see e.g. [4]), i.e.

$$\hat{u}(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{z}_j||_2), \tag{3}$$

where c_j denote unknown real coefficients, $||\cdot||_2$ is the Euclidean norm, and $\phi_\varepsilon : [0, \infty) \to \mathbb{R}$ is some RBF depending on a positive *shape parameter* ε such that

$$\phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{z}||_2) = \phi(\varepsilon||\boldsymbol{x} - \boldsymbol{z}||_2), \qquad \forall\, \boldsymbol{x}, \boldsymbol{z} \in \Omega.$$

As an example, globally supported RBFs that are commonly used for solving PDEs are listed below along with their smoothness degrees (see [11]):

$$\phi_\varepsilon(r) = (1 + \varepsilon^2 r^2)^{-1/2}, \qquad \text{Inverse MultiQuadric } C^\infty,$$
$$\phi_\varepsilon(r) = \exp(-\varepsilon r)(\varepsilon^3 r^3 + 6\varepsilon^2 r^2 + 15\varepsilon r + 15), \qquad \text{Matérn } C^6.$$

Note that the value of ε significantly affects stability and accuracy of a RBF method. In particular, in (3) the accuracy is typically high (low) when ε is small (large), but the ill-conditioning is severe (acceptable). For further details, see [4].

In (3) we distinguish between *centers* $Z_N = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ and *collocation points* $X_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \Omega$. Even though such sets of points can formally

be distinct, for the following discussion we assume that $Z_N = X_N$ (see [4]). Then, for our purposes we split the set X_N in the two subsets X_{N_I} of interior points and X_{N_B} of boundary points, with $X_N = X_{N_I} \cup X_{N_B}$, N_I and N_B indicating the number of interior and boundary points, respectively.

When matching the differential equation (1) and the boundary conditions (2) at the collocation points X_N, we get the collocation system of linear equations

$$A\boldsymbol{c} = \boldsymbol{u},$$

where A is the collocation matrix

$$A = \begin{bmatrix} \hat{A}_{\mathcal{L}} \\ \hat{A} \end{bmatrix}, \tag{4}$$

whose two blocks in (4) are given by

$$\begin{aligned} (\hat{A}_{\mathcal{L}})_{ij} &= \mathcal{L}\phi(||\boldsymbol{x}_i - \boldsymbol{z}_j||_2), \qquad \boldsymbol{x}_i \in X_{N_I},\ \ \boldsymbol{z}_j \in Z_N, \\ \hat{A}_{ij} &= \phi(||\boldsymbol{x}_i - \boldsymbol{z}_j||_2), \qquad \boldsymbol{x}_i \in X_{N_B},\ \ \boldsymbol{z}_j \in Z_N, \end{aligned}$$

while $\boldsymbol{u}$ is the vector of entries

$$u_i = \begin{cases} f(\boldsymbol{x}_i), & \boldsymbol{x}_i \in X_{N_I}, \\ g(\boldsymbol{x}_i), & \boldsymbol{x}_i \in X_{N_B}. \end{cases}$$

Kansa's approach is known to be a nonsymmetric collocation method. Theoretical analysis and further considerations on this popular RBF method can be found, for instance, in [5,8].

3 Adaptive LOOCV-based Scheme

3.1 Basics on LOOCV

The idea of LOOCV for Kansa's method can be depicted in the following way. At first, the data are split into two distinct sets: a *training set* $\{u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_{k-1}), u(\boldsymbol{x}_{k+1}), \ldots, u(\boldsymbol{x}_N)\}$, and a *validation set* that is merely made of the single value $u(\boldsymbol{x}_k)$, i.e. the one left out when generating the training set [4].

Given an index $k \in \{1, \ldots, N\}$ and a fixed value of ε, the partial RBF approximant is given by

$$\hat{u}^{[k]}(\boldsymbol{x}) = \sum_{j=1,\ j\neq k}^{N} c_j^{[k]} \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{z}_j||_2),$$

whose coefficients $c_j^{[k]}$ are found by collocating the training data, that is

$$\begin{aligned} \mathcal{L}\hat{u}^{[k]}(\boldsymbol{x}_i) &= f(\boldsymbol{x}_i), \qquad \boldsymbol{x}_i \in X_{N_I}, \\ \hat{u}^{[k]}(\boldsymbol{x}_i) &= g(\boldsymbol{x}_i), \qquad \boldsymbol{x}_i \in X_{N_B}, \end{aligned} \quad \text{for } i = 1, \ldots, k-1, k+1, \ldots, N.$$

In order to give a measure of the quality of this attempt, we define the absolute error

$$e_k = |u(\boldsymbol{x}_k) - \hat{u}^{[k]}(\boldsymbol{x}_k)|, \tag{5}$$

at that validation point $\boldsymbol{x}_k$, which is not used to determine the approximant. Now, if we compute the error in (5), for all $k = 1, \ldots, N$, we get a vector $\boldsymbol{e} = (e_1, \ldots, e_N)^T$, which can be viewed as an error indicator to identify the regions which require to be refined by adding any points in the neighborhood of the selected point. However, instead of using (5), the computation of the error components can be done in a more efficient way without solving N collocation problems, each of size $(N-1) \times (N-1)$. In fact, in [10] Rippa shows that the computation of the error terms can be simplified to a single formula. The rule (5) can thus be rewritten as

$$e_k = \left| \frac{c_k}{A^{-1}_{kk}} \right|, \qquad k = 1, \ldots, N, \tag{6}$$

where c_k is the k-th coefficient of the full approximate solution (3) and A^{-1}_{kk} is the k-th diagonal element of the inverse of the corresponding $N \times N$ collocation matrix A in (4).

3.2 Error Indicator and Refinement via LOOCV

At the beginning of our adaptive scheme, we define an initial set $X^1_{N^{(1)}} \equiv X_N = \{\boldsymbol{x}^{(1)}_1, \ldots, \boldsymbol{x}^{(1)}_{N^{(1)}}\}$ of grid collocation points. It is then split into two subsets, i.e. the set $X^1_{N^{(j)}_I} = \{\boldsymbol{x}^{(j)}_1, \ldots, \boldsymbol{x}^{(j)}_{N^{(j)}_I}\}$ of interior points, and the set $X^1_{N^{(j)}_B} = \{\boldsymbol{x}^{(j)}_1, \ldots, \boldsymbol{x}^{(j)}_{N^{(j)}_B}\}$ of boundary points, where $j = 1, 2, \ldots$ identifies the iteration of the adaptive algorithm. Observe that the superscript ^{1} above denotes the first phase of our scheme, while the subscript $_{N^{(j)}}$ defines the number of collocation points in the j-th iteration of that same phase.

Then, for a fixed tolerance $\tau > 0$ in our adaptive process, from (6) we can iteratively define an error indicator via LOOCV given by

$$e^{(j)}_k = \left| \frac{c_k}{A^{-1}_{kk}} \right|, \qquad k = 1, \ldots, N^{(j)}, \qquad j = 1, 2, \ldots, \tag{7}$$

where in the absolute value argument any reference to the iteration is omitted to avoid confusion in the notation. If the error indicator (7) is such that $e^{(j)}_k > \tau$, then a refinement is applied in the neighborhood of $\boldsymbol{x}_k$. However, in order to do that, first we have to compute the so-called *separation distance*

$$q_{X^1_{N^{(j)}}} = \frac{1}{2} \min_{u \neq v} ||\boldsymbol{x}^{(j)}_u - \boldsymbol{x}^{(j)}_v||_2, \qquad \boldsymbol{x}^{(j)}_u, \boldsymbol{x}^{(j)}_v \in X^1_{N^{(j)}}, \qquad j = 1, 2, \ldots, \tag{8}$$

and then we sum up or subtract the quantity in (8) to one (both) coordinate(s) of the point $\boldsymbol{x}_k$. In particular, defining explicitly the coordinates of $\boldsymbol{x}_k$, i.e. setting

$\boldsymbol{x}_k = (\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2})$, the refinement strategy consists in the addition of four points around $\boldsymbol{x}_k$. More precisely, the coordinates of these four points are $(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2} + q_{X^1_{N(j)}})$, $(\boldsymbol{x}_{k,1} - q_{X^1_{N(j)}}, \boldsymbol{x}_{k,2})$, $(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2} - q_{X^1_{N(j)}})$ and $(\boldsymbol{x}_{k,1} + q_{X^1_{N(j)}}, \boldsymbol{x}_{k,2})$. The refinement procedure stops when all components of error terms in (7) are less than or equal to the tolerance τ.

3.3 Further Adaptive Refinement

At the second phase of the algorithm we further refine the collocation points based on other information coming from a new error estimate. Here, this adaptive refinement is a sort of extension of the residual subsampling method given in [3], and later modified for RBF partition of unity collocation [1]. Basically, our procedure is obtained by means of an efficient combination between an error indicator and a refinement technique, which consists in solving, estimating and finally adding or removing adaptively any points in the areas selected by the process.

Therefore, we start from the output of the previous phase described in Subsection 3.2 and re-name that final set as $X^2_{N(1)}$, where the superscript ^{2} denotes the second stage of our adaptive process. Then, for $k = 1, 2, \ldots$, we compute iteratively two solutions of the form (3), called $\hat{u}$ and $\hat{u}^a$, respectively. In particular, $\hat{u}$ is found by considering the set $X^2_{N(k)}$ of collocation points, while $\hat{u}^a$ is obtained by taking the same set $X^2_{N(k)}$ with additional boundary points outside the domain. The resulting set is denoted as $X^{2,a}_{N(k)}$. Further details on this strategy of adding points, which is here used to get two approximate RBF solutions, can be found in [4, Chapter 39].

After defining a set $Y^{(k)} = \{\boldsymbol{y}^{(k)}_1, \ldots, \boldsymbol{y}^{(k)}_{N(k)}\}$ of check points, we validate the results via the error indicator defined by

$$E^{(k)}_i = |\hat{u}(\boldsymbol{y}^{(k)}_i) - \hat{u}^a(\boldsymbol{y}^{(k)}_i)|, \qquad \boldsymbol{y}^{(k)}_i \in Y^{(k)}. \tag{9}$$

At the moment we fix two thresholds τ_{low} and τ_{upp}, such that $0 < \tau_{low} < \tau_{upp}$, which allow us to refine the set of discretization points when the estimate indicator (9) does not give precise enough result, or coarsen the set of discretization points if the level of accuracy achieved is under the tolerance τ_{low}. This process thus leads to an addition or removal of points, thus making this scheme adaptive. The iterative procedure stops once the refinement process is completed.

4 Numerical Results

In this section we show some results derived from application of our adaptive algorithm, which is implemented in MATLAB. All results are carried out on a laptop with an Intel(R) Core(TM) i7-6500U CPU 2.50 GHz processor and 4GB RAM.

In the following we restrict our attention on solution of some 2D elliptic PDE problems with Kansa's collocation method. More precisely, we consider some

Poisson problems, taking the Laplace operator $\mathcal{L} = -\Delta$ in (1). To analyze the performance of our adaptive scheme, we take two test problems defined on the domain $\Omega = [0,1]^2$ (see [1,7]). The analytic solutions of such Poisson problems are given by

$$u_1(x_1, x_2) = \frac{1}{20}\exp(4x_1)\cos(2x_1 + x_2),$$
$$u_2(x_1, x_2) = \sinh(0.3(4x_1 - 4)\sin(8x_2 - 4)\exp(-(4x_1 - 2.1)^4)),$$

while their graphical representation is shown in Figure 1.

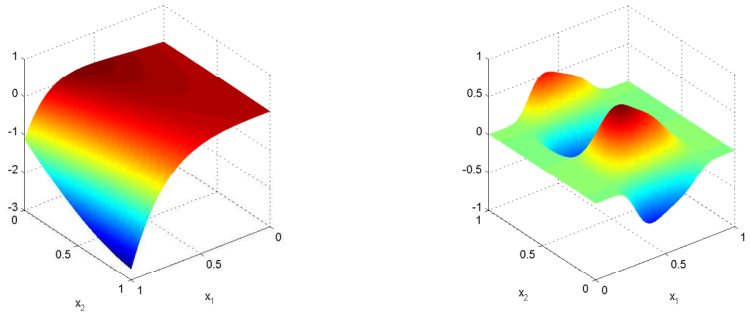

Fig. 1. Exact solutions of Poisson problems u_1 (left) and u_2 (right).

Moreover, we remark that in the second stage of the algorithm we used Halton points [4] as check points. It is however obvious that the choice of these validation points is absolutely arbitrary and other possible distributions can be used.

In our tests we illustrate the performance of the adaptive algorithm applied to Kansa's approach by using RBFs of different smoothness as Inverse Multi-Quadric C^∞ (IMQ) and Matérn C^6 (M6) functions (see Section 2). In these experiments we begin the iterative process by considering $N = 225$ grid collocation points, consisting of $N_I = 169$ interior points and $N_B = 56$ boundary points. In order to investigate the accuracy of the adaptive scheme, we compute the Root Mean Square Error (RMSE), i.e.

$$\text{RMSE} = \sqrt{\frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} |u(\boldsymbol{\xi}_i) - \hat{u}(\boldsymbol{\xi}_i)|^2},$$

where the $\boldsymbol{\xi}_i$, $i = 1, \ldots, N_{eval}$, constitute a set of grid evaluation points and $N_{eval} = 40 \times 40$. Further, to analyze the stability of the numerical method, we evaluate the Condition Number (CN) of the collocation matrix A in (4) by making use of the MATLAB command `cond`.

After carrying out a preliminary analysis of the algorithm behavior with different values of ε in the interval $[1, 6]$, in Table 1 we show a summary of all results obtained by using IMQ and M6 with $\varepsilon = 4$, also indicating the total number N_{fin} of collocation points obtained to achieve the final result. While the value of τ associated with the LOOCV is related to the PDE problem, the values of $(\tau_{low}, \tau_{upp}) = (10^{-8}, 10^{-4})$ are kept fixed. In Figure 2 we show some graphs with the final configurations of discretization points, while Figure 3 shows some plots of the absolute error for RBF solution, i.e. $|u(\boldsymbol{\xi}_i) - \hat{u}(\boldsymbol{\xi}_i)|$, $i = 1, \ldots, N_{eval}$, computed on the grid of $N_{eval} = 40 \times 40$ evaluation points.

Problem	RBF	N_{fin}	RMSE	CN	no. iter	τ
u_1	M6	547	9.22e−6	7.57e+11	6	0.05
u_1	IMQ	810	4.54e−5	5.88e+15	9	0.05
u_2	M6	1086	2.97e−5	3.91e+11	6	0.1
u_2	IMQ	695	2.23e−5	3.50e+14	4	0.1

Table 1. Results obtained by starting from $N = 225$ grid collocation points and using various RBFs with $\varepsilon = 4$ and $(\tau_{low}, \tau_{upp}) = (10^{-8}, 10^{-4})$.

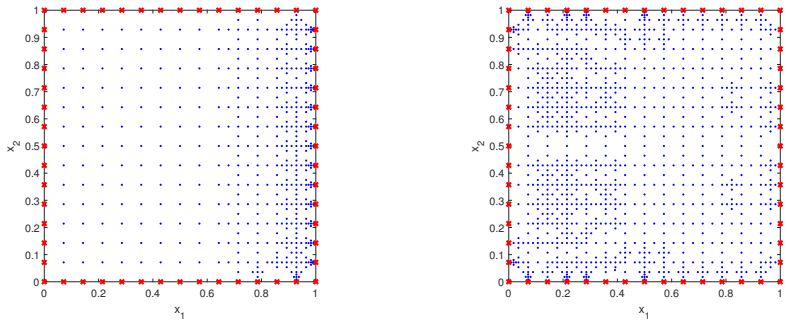

Fig. 2. Final discretization points for u_1 (left) and u_2 (right) using M6 with $\varepsilon = 4$.

From an analysis of the numerical results, we can note that our adaptive refinement scheme works well for both differential problems. The algorithm identifies the areas of Ω with significant variations, increasing the number of collocation points only in those domain parts in which the accuracy is not enough. As regards the stability, we remark that the CN assumes values around 10^{+11}–10^{+15}. Even though RBF-based methods can suffer from severe ill-conditioning (see e.g. [4]), we observe as a good refinement strategy allows us to control it, thus avoiding the number of discretization points increasing by too much. Finally, in Table 1 we indicate the number of iteration required, observing that our iterative algorithm completes its refinement process in only a few seconds.

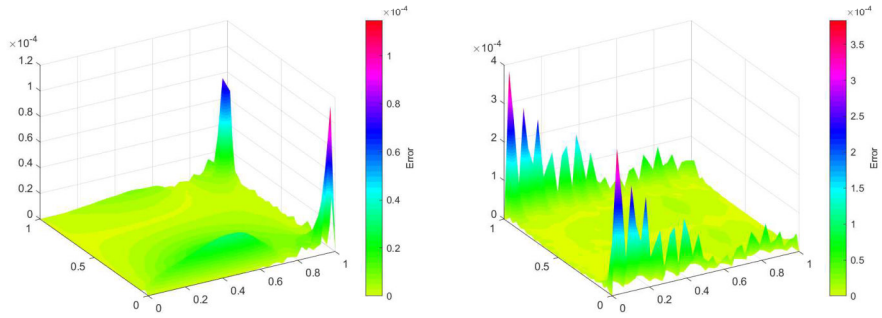

Fig. 3. Absolute error for u_1 (left) and u_2 (right) using M6 with $\varepsilon = 4$.

References

1. Cavoretto, R., De Rossi, A.: Adaptive meshless refinement schemes for RBF-PUM collocation. Appl. Math. Lett. **90**, 131–138 (2019)
2. Chen, W., Fu, Z.-J., Chen, C.S.: Recent Advances on Radial Basis Function Collocation Methods. Springer Briefs in Applied Science and Technology, Springer, Heidelberg (2014)
3. Driscoll, T.A., Heryudono, A.R.H.: Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. Comput. Math. Appl. **53**, 927–939 (2007)
4. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB. Interdisciplinary Mathematical Sciences, vol. 6, World Scientific Publishing Co., Singapore (2007)
5. Hon, Y.C., Schaback, R.: On unsymmetric collocation by radial basis functions. Appl. Math. Comput. **119**, 177–186 (2001)
6. Kansa, E.J.: Multiquadrics–A scattered data approximation scheme with applications to computational fluid-dynamics–II solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput. Math. Appl. **19**, 147–161 (1990)
7. Larsson, E., Shcherbakov, V., Heryudono, A.: A least squares radial basis function partition of unity method for solving PDEs. SIAM J. Sci. Comput. **39**, A2538–A2563 (2017)
8. Lee, C.-F., Ling, L., Schaback, R.: On convergent numerical algorithms for unsymmetric collocation. Adv. Comput. Math. **30**, 339–354 (2009)
9. Oanh, D.T., Davydov, O., Phu, H.X.: Adaptive RBF-FD method for elliptic problems with point singularities in 2D. Appl. Math. Comput. **313**, 474–497 (2017)
10. Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. Adv. Comput. Math. **11**, 193–210 (1999)
11. Wendland, H.: Scattered Data Approximation. Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, Cambridge (2005)
12. Yang, J., Liu, X., Wen, P.H.: The local Kansa's method for solving Berger equation. Eng. Anal. Bound. Elem. **57**, 16–22 (2015)