ARTICLE TEMPLATE

Solving nonlinear elliptic PDEs in 2D and 3D using polyharmonic splines and low-degree of polynomials

Kalani Rubasinghe[a] and Guangming Yao[a] and Wen Li[b] and Gantumur Tsogtgerel[c]

[a]Department of Mathematics, Clarkson University, USA
[b]Departments of Mathematics, University of California, Los Angeles, USA
[c]Department of Mathematics and Statistics, McGill University, Canada

ARTICLE HISTORY
Compiled June 15, 2023

ABSTRACT
In this paper, the improved localized method of approximated particular solutions (ILMAPS) using polyharmonic splines (PHS) together with a low-degree of polynomial basis is used to approximate solutions of various nonlinear elliptic Partial Differential Equations (PDEs). The method is completely meshfree, and it uses a radial basis function (RBF) that has no shape parameters. The discretization process is done through a simple collocation technique on a set of points in the local domain of influence. Resulted system of nonlinear algebraic equations is solved by the Picard method.

The performance of the proposed method is tested on various nonlinear elliptical problems, including the Poisson-type problems in 2D and 3D with constant or variable coefficients on rectangular or irregular domains and the Poisson-Boltzmann equation with Dirichlet boundary conditions or mixed boundary conditions. The effect of domain shapes in 2D and 3D, types of boundary conditions, and degrees of polyharmonic splines, and order of polynomial basis are examined. The performance of the method is compared with other bases such as multiquadrics (MQ) basis functions, and with results reported in the literature (method of particular solutions using polynomials). The numerical experiments suggest that ILMAPS with polyharmonic splines yields considerably superior accuracy than other RBFs as well as other approaches reported in the literature for solving nonlinear elliptic PDEs.

KEYWORDS
Nonlinear elliptic equation, LMAPS, radial basis functions, polyharmonic splines, multiquadradrics

1. Introduction

Mesh-based methods, such as finite difference method (FDM) [21, 29], finite volume method (FVM) [8, 23] and finite element method (FEM) [4] are commonly used numerical methods in computational sciences and engineering. Meshfree methods, on the other hand, have been developed and effectively applied to solve many problems in science and engineering during the last two decades [22, 24–26, 35]. Most of those meshfree methods are collocation methods where scattered nodes in the domain and on the boundaries of the domain are used to establish an algebraic system of equations. Strong form methods account for a large proportion of the meshfree methods,

Contact G. Yao gyao@clarkson.edu

such as the vortex method [1], general FDM [28], the meshfree collocation method [15], the method of fundamental solution [14], the method of fundamental solution-Method of particular solution [16], the method of particular solution [3], the localized method of particular solutions [34], and etc. The complex pre-processing is avoided, and the formulation procedure is straightforward. These strong form meshfree methods have some attractive features: no mesh or numerical integration is needed during the discretization process, they are simple to implement, and they are efficient in comparison with mesh-based methods.

Radial basis functions (RBFs) are used in the discretization process in the strong form meshfree method, to create shape functions. Any functions whose values depend only on the distance from a fixed point (also called a center), x_c, i.e.,

$$\phi(\mathbf{x}) = \phi(||\mathbf{x} - \mathbf{x}_c||) = \phi(r), \tag{1}$$

are called radial basis functions (RBFs), where $||\cdot||$ denotes the Euclidean norm, $r = ||\mathbf{x}-\mathbf{x}_c||$. Thus, discretization using RBFs reduces problems in high-dimensional spaces to one-dimensional problems. Some commonly used RBFs are listed in Table 1, which include Gaussian, multiquadrics (MQ), Matern, polyharmonic splines (PHS) and etc. Note $K_i, i = 2, 3$ denote the modified Bessel functions of the second kind of order two and three, respectively. Among those, all but polyharmonic splines are positive definite RBFs, where the collocation matrix constructed from distinct collocation points is invertible. Thus, all but polyharmonic splines have been widely used in applications and numerical simulations of solutions to ordinary and partial differential equations. However, the strong form meshfree methods using those RBFs are somewhat sensitive to the choice of shape parameters of RBFs and can be unstable due to ill-conditioning issues.

Table 1.: Commonly used RBFs.

RBF	Formulation	Shape parameter
Gaussian	$\phi(r) = exp(-cr^2),\ c > 0$	c
multiquadric (MQ)	$\phi(r) = \sqrt{r^2 + c^2}, c > 0$	c
inverse multiquadric (IMQ)	$\phi(r) = 1/\sqrt{r^2 + c^2},\ c > 0$	c
Matern	$\phi(r) = (cr)^m K_m(cr),\ m > 0$	c
Polyharmonic Splines (PHS) in 2D	$\phi(r) = r^{2k} ln(r),\ k \in \mathbb{N}^+$	None
Polyharmonic Splines (PHS) in 3D	$\phi(r) = r^{2k-1},\ k \in \mathbb{N}^+$	None

The localized method of particular solutions (LMAPS) was first introduced in 2010 [34]. The performance of LMAPS on linear PDEs has been studied using various types of PDEs for past years, including two-dimensional velocity–vorticity formulation of the Navier–Stokes equations [9] using implicit Euler method and the Newton's method, 3D nonlinear Schrödinger equations [20], biharmonic equations [17], unsteady Navier–Stokes problem [36], two-dimensional nonlinear sine-Gordon equation [27], Cole–Hopf transformation for multi-dimensional Burgers equations [19], near-singular two-and three-dimensional problems [33], and etc.

To make these types of meshfree methods more efficient and robust and to achieve even higher accuracy, the LMAPS method is improved by utilizing the PHS RBF for solving linear elliptic PDEs [31]. The main differences between the original LMAPS and

the Improved LMAPS is that the ILMAPS uses polyharmonic splines and a low degree of polynomial basis to approximate solutions whereas original LMAPS is limited to Gaussian, MQ, or Matern RBFs. Such improvement avoided the difficulty of searching for the optimal shape parameter, accuracy is improved simply by increasing the order of polyharmonic splines or polynomial basis, or the number of interpolation points, and better stability with the use of additional polynomial basis. To our surprise, the method is extremely accurate and efficient, in comparison to all other commonly used RBFs. To improve the stability, a basis of low degree of polynomials is added to create the shape functions. The basis of a low degree of polynomials guarantees the non-singularity of the collocation matrices. The improved LMAPS (ILMAPS) is then applied to solve linear elliptic equations with mixed boundary conditions on scattered data [18].

The effect of boundary conditions on the global method of approximated particular solutions (MAPS) was studied in [12]. In [32], MAPS with PHS has successfully been used for solving nonlinear PDEs. In 2017, Dangal *et al* proposed the method of particular solutions using polynomials only (without the RBFs). The method is proved to be efficient and accurate [6].

Nonlinear partial differential equations are nowadays very popular as many real-world problems, ranging from gravitation to fluid dynamics are modeled by these equations. There is no general method to solve such problems, especially when dealing with irregular geometries and complex boundary conditions. Therefore, a numerical algorithm that is aiming for solving general nonlinear elliptic PDEs with complicated features such as domains or boundary conditions is needed.

In this paper, the improved localized method of approximated particular solutions (ILMAPS) using PHS together with a basis of low degree of polynomials is used to approximate solutions of various nonlinear PDEs, particularly to the elliptic type PDEs. We used the Picard method for the nonlinear iteration in all examples which will be discussed in Section 3. The effect of domain shapes, the complexity of the nonlinear terms in the elliptic equations in 2D and 3D, types of boundary conditions including Dirichlet boundary condition or mixed Dirichlet and Neumann boundary conditions, the order of PHS, and the degree of polynomial basis are examined. The performance of the proposed method is compared with other bases such as multiquadrics basis (MQ) functions and other reported results in the literature. We show that the proposed ILMAPS method can avoid some of the ill-conditioning issues and is more accurate and efficient than the original method, especially for the nonlinear elliptic PDEs using Neumann boundary conditions, whereas the original LMAPS was not able to find accurate solutions when dealing with Neumann boundary conditions.

2. Improved localized method of approximated particular solutions (ILMAPS)

Let us consider the following elliptic PDEs of the form

$$\mathcal{D}u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2}$$

$$\mathcal{B}u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \tag{3}$$

where $\mathcal{D}$ and $\mathcal{B}$ are differential operators, f and g are known functions, $\Omega \in \mathbb{R}^d, d = 2, 3$ is a bounded and closed domain with boundary $\partial\Omega$. This paper focuses on nonlinear differential operators for $\mathcal{D}$ and $\mathcal{B}$.

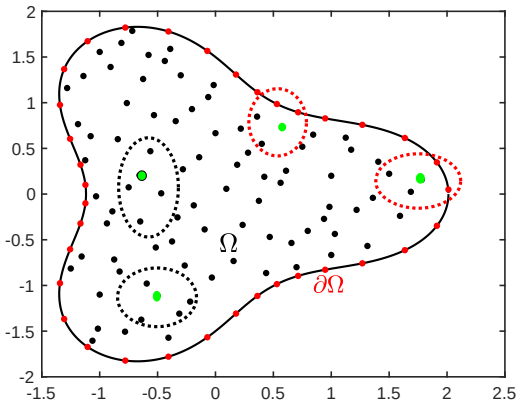

Figure 1.: The random node arrangement on an irregular domain and the schematics of the five-node local domains of influence in the interior and at the near boundary.

Let $\mathcal{B}_m^d$ be the set of d-variant polynomials of degree up to m and $\{p\}_{l=1}^w$ be a basis of $\mathcal{B}_m^d$ where

$$w = \binom{m+d}{d} = \begin{cases} \frac{1}{2}(m+2)(m+1), & \text{in } \mathbb{R}^2, \\ \frac{1}{6}(m+1)(m+2)(m+3), & \text{in } \mathbb{R}^3, \end{cases}$$

is the dimension of the polynomial basis. Let $\{\mathbf{x}_i\}_{i=1}^N \in \Omega \cup \partial\Omega$ be the spatial discretization points, where the first n_i points are the interior points in Ω, followed by n_b boundary points on $\partial\Omega$ such that $n_i + n_b = N$.

In ILMAPS, the solution of PDEs can be approximated on a set of radial basis functions. First of all, for each $\mathbf{x}_i \in \Omega$, a local domain of influence, Ω_i needs to be created. This can be done through a kd-tree search or knn search for a large amount of data. Figure 1 is a schematic showing five-node local domains of influence. Let n be the number of points in the influence domains, and $\Omega_i = \{\mathbf{x}_j^{[i]}\}_{j=1}^n$. Let $\phi(r)$ be the PHS of order k and $\{p_l\}_{l=1}^w$ be the polynomial basis of order up to m. Then the solution to (2)-(3) can be approximated by particular solutions and polynomials in the following way:

$$u(\mathbf{x}_i) \approx \hat{u}(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_j \Phi(||\mathbf{x}_i - \mathbf{x}_j^{[i]}||) + \sum_{l=1}^{w} \alpha_{n+l} p_l(\mathbf{x}_i); \quad \mathbf{x}_i \in \Omega \tag{4}$$

where $\mathbf{x}_j^{[i]}, j = 1, \cdots, n \in \Omega_i$ are points located inside the local domain of influence of $\mathbf{x}_i$, $\{\alpha_j\}$ is the undetermined coefficients, Φ is a particular solution with respect to ϕ and differential operator $\mathcal{D}$, which is also a RBF, i.e

$$\mathcal{D}\Phi(r) = \phi(r),$$

and the augmented polynomial basis is as follows: for any $0 \leq l \leq w$,

$$p_l(\mathbf{x}) = \begin{cases} x^{l-j}y^j, & 0 \leq j \leq i,\ 0 \leq i \leq m, \ \text{ in } \mathbb{R}^2, \\ x^{l-j-k}y^j z^k, & 0 \leq k \leq i-j,\ 0 \leq j \leq i,\ 0 \leq i \leq m, \ \text{ in } \mathbb{R}^3, \end{cases}$$

where m is the highest degree of polynomials in the polynomial basis. Particular solutions for various differential operators associated with commonly used RBFs have already been derived in [30]. The closed form particular solutions

$$\Delta\Phi(r) = \phi(r) = r^{2k}\ln(r)$$

in $\mathbb{R}^2$ is

$$\Phi(r) = \frac{r^{(2k+2)}\ln(r)}{4(k+1)^2} - \frac{r^{(2k+2)}}{4(k+1)^3}. \tag{5}$$

Since there are w additional degrees of freedoms in (4), standard polynomial insolvency constraint [11, 32] must be applied. Thus, the collocation technique on a local domain of $\mathbf{x}$ resulted in the following linear system:

$$\sum_{j=1}^{n} \alpha_j \Phi(\|\mathbf{x}_k^{[i]} - \mathbf{x}_j^{[i]}\|) + \sum_{l=1}^{w} \alpha_{n+l} p_l(\mathbf{x}_k^{[i]}) = \hat{u}(\mathbf{x}_k^{[i]}), \quad k = 1, 2, \ldots, n, \tag{6}$$

$$\sum_{j=1}^{n} \alpha_j p_l(\mathbf{x}_k^{[i]}) = 0, \quad l = 1, 2, \ldots, w. \tag{7}$$

Note that (6)–(7) is a linear system of equations with $n + w$ coefficients to be determined. Let's denote the coefficient matrix in the first term in (6) as $\mathbf{\Phi}_{nn}$, and the second term as $\mathbf{P}_{nw}$. Note that the matrix $\mathbf{\Phi}_{nn}$ is a symmetric matrix of size $n \times n$. Then the above system can be represented in block matrix form

$$\begin{bmatrix} \mathbf{\Phi}_{nn} & \mathbf{P}_{nw} \\ \mathbf{P}_{nw}^T & \mathbf{0}_{ww} \end{bmatrix} \boldsymbol{\alpha}_{\boldsymbol{n+w}} = \begin{bmatrix} \hat{\mathbf{u}}_n \\ \mathbf{0}_w \end{bmatrix}, \tag{8}$$

where

$$\hat{\mathbf{u}}_n = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \cdots, \hat{u}(\mathbf{x}_n)]^T, \ \boldsymbol{\alpha}_{\boldsymbol{n+w}} = [\alpha_1, \alpha_2, \cdots, \alpha_{n+w}]^T,$$

and $\mathbf{0}_w$ is a zero matrix of size $w \times w$. Denote the coefficient matrix in (8) by $\mathbf{\Phi}_{n+w}$, and the right-hand side of the (8) by $\hat{\mathbf{u}}_{n+w}$. Then the system can be rewritten as

$$\mathbf{\Phi}_{n+w}\boldsymbol{\alpha}_{\boldsymbol{n+w}} = \hat{\mathbf{u}}_{n+w}. \tag{9}$$

Since both weights $\boldsymbol{\alpha}_{\boldsymbol{n+w}}$ and approximated solutions $\hat{\mathbf{u}}_n$ are unknown, we can interchange those two vectors in the system above:

$$\mathbf{\Phi}_{n+w}^{-1}\hat{\mathbf{u}}_{n+w} = \boldsymbol{\alpha}_{\boldsymbol{n+w}}. \tag{10}$$

Note that the matrix $\mathbf{\Phi}_{n+w}$ is nonsingular if the nodes inside Ω_i are distinct. The unknown coefficient vector $\boldsymbol{\alpha}_{n+w}$ in (8) can be reformulated as in (10), although the inverse matrix $\mathbf{\Phi}_{n+w}^{-1}$ exists but we never computed it directly. Details can be found below:

- Plug (9) into (2)-(3), we have

$$\mathcal{D}\mathbf{\Phi}_{n+w}\boldsymbol{\alpha}_{\boldsymbol{n+w}} = f(\mathbf{x}_i), \quad \mathbf{x}_i \in \Omega, \tag{11}$$
$$\mathcal{B}\mathbf{\Phi}_{n+w}\boldsymbol{\alpha}_{\boldsymbol{n+w}} = g(\mathbf{x}_i), \quad \mathbf{x}_i \in \partial\Omega, \tag{12}$$

- Plug (10) into the system above, we have

$$\left((\mathcal{D}\mathbf{\Phi}_{n+w})\,\mathbf{\Phi}_{n+w}^{-1}\right)\hat{\mathbf{u}}_{n+w} = f(\mathbf{x}_i), \quad \mathbf{x}_i \in \Omega, \tag{13}$$
$$\left((\mathcal{B}\mathbf{\Phi}_{n+w})\,\mathbf{\Phi}_{n+w}^{-1}\right)\hat{\mathbf{u}}_{n+w} = g(\mathbf{x}_i), \quad \mathbf{x}_i \in \partial\Omega. \tag{14}$$

- Coefficients of unknown approximations $\hat{\mathbf{u}}_{n+w}$ in (13) or (14), denoted by A, is a row vector of size $(n+w)$. This can be obtained by solving the following system, for (13),

$$\mathbf{\Phi}_{n+w}A^T = (\mathcal{D}\mathbf{\Phi}_{n+w})^T. \tag{15}$$

This is why we say that the inverse of small matrices of the size of $(n+w)\times(n+w)$ were never computed directly. It was done by solving a small linear system. Since there were N collocation matrices in the spatial discretization, we will need to solve N systems, which resulted in a large system, (13)-(14), of N nonlinear equations with N unknowns $\hat{u}(x_i) \approx u(\mathbf{x}_i), \mathbf{x}_i \in \Omega \cup \partial\Omega$. This $N \times N$ sparse nonlinear system of equations can be solved by an efficient sparse nonlinear solver. To our surprise, the simple Picard method is already sufficient. This will be discussed in the next section.

3. Nonlinear Solver–Picard Method

If the differential operator $\mathcal{D}$ is nonlinear, a direct Picard method is used to solve the nonlinear system of algebraic equations (13)-(14). For our simplicity, we denote the nonlinear system of N equations with N unknowns $\hat{u}(\mathbf{x}_i) \approx u(\mathbf{x}_i), \mathbf{x}_i \in \Omega \cup \partial\Omega$ by

$$\mathbf{A}(\hat{\mathbf{u}})\hat{\mathbf{u}} = \mathbf{b}, \tag{16}$$

where $\mathbf{A}(\hat{\mathbf{u}})$ as an $N \times N$ matrix function of $\hat{u}$, $\mathbf{b}$ as a vector function, and $\hat{\mathbf{u}} = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \cdots, \hat{u}(\mathbf{x}_N)]^T$.

Example 3.1. *Let the differential operator*

$$\mathcal{D}u = \Delta u - u\frac{\partial u}{\partial x} + u^2, \qquad \mathcal{B} = \mathbf{I}.$$

The resulting nonlinear system by ILMAPS would be

$$\left(\left(\Delta\mathbf{\Phi}_{n+w}\right)\mathbf{\Phi}_{n+w}^{-1}\right)\hat{\mathbf{u}}_{n+w} - \hat{u}_i\left(\left(\frac{\partial}{\partial x}\mathbf{\Phi}_{n+w}\right)\mathbf{\Phi}_{n+w}^{-1}\right)\hat{\mathbf{u}}_{n+w} + \hat{u}_i^2 = f(\mathbf{x}_i), \quad \mathbf{x}_i \in \Omega,$$
$$\hat{u}_i = g(\mathbf{x}_i), \quad \mathbf{x}_i \in \partial\Omega,$$

where $\hat{u}_i \approx u(\mathbf{x}_i)$ is the approximated solution at $\mathbf{x}_i$. The above system can be represented as a nonlinear system of the following form:

$$\mathbf{A}(\hat{\mathbf{u}})\hat{\mathbf{u}} = \mathbf{b}(\hat{\mathbf{u}}).$$

We can linearize the product $\mathbf{A}(\hat{\mathbf{u}})\hat{\mathbf{u}}$ to $\mathbf{A}(\hat{\mathbf{u}}_i)\hat{\mathbf{u}}$ and $\mathbf{b}(\hat{\mathbf{u}})$ as $\mathbf{b}(\hat{\mathbf{u}}_i)$. That is, we use the most previously computed approximation in $\mathbf{A}$ and $\mathbf{b}$ to arrive at a linear system for $\hat{\mathbf{u}}$. Let the initial guess $\hat{\mathbf{u}}_0 = \mathbf{0}$. We construct a sequence $\hat{\mathbf{u}}_i, i = 1, 2, 3, \ldots$ by solving the following linear system:

$$\mathbf{A}(\hat{\mathbf{u}}_i)\hat{\mathbf{u}}_{i+1} = \mathbf{b}(\hat{\mathbf{u}}_i), i = 0, 1, 2, \cdots .$$

The algorithm can be found below:

Algorithm 1: The Picard Method for solving (16).

Data: Set a small positive tolerance ϵ;
Set large positive integer tolerance TOL;
Result: $\hat{\mathbf{u}}_i$

$\hat{\mathbf{u}}_0$ initial guess, $i = 0$;
$\hat{\mathbf{u}}_1 = 10\epsilon\hat{\mathbf{u}}_0$;
while $\|\hat{\mathbf{u}}_i - \hat{\mathbf{u}}_{i+1}\| > \epsilon$ **do**
  $i = i + 1$;
  if $i < TOL$ **then**
    Solve $\mathbf{A}((\hat{\mathbf{u}}_i))\hat{\mathbf{u}}_{i+1} = \mathbf{b}(\hat{\mathbf{u}}_i)$;
  else
end

Then $\hat{\mathbf{u}}_i$ at the final iteration is the approximated solution.

4. Numerical Results

In this section, the improved localized method of approximated particular solutions (ILMAPS) using polyharmonic splines (PHS) together with a basis of low degree of polynomials is used to approximate solutions of various nonlinear Poisson-type elliptic PDEs on regular and irregular domains in 2D or 3D. Recall the following notations:

- n_i: the number of collocation points in the domain Ω
- n_b: the number of collocation points on the boundary $\partial\Omega$
- $N = n_i + n_b$: the total number of collocation points
- n: the number of points in the local domain of influence
- m: the degree of highest order polynomials
- k: the order of PHS.

Uniformly distributed interior points are used in the computational experiments, together with points on the boundaries with equal spaces along the boundary. The root mean squared error (ϵ_{rms}) and the maximum absolute error (ϵ_∞) are used to measure the accuracy of our approximated solutions. They are defined as follows:

$$\epsilon_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{u}_i - u_i)}, \qquad \epsilon_\infty = \max_{i=1}^{N}|\hat{u}_i - u_i| \tag{17}$$

where $\hat{u}_i \approx u_i = u(\mathbf{x}_i)$. The Picard method is used for nonlinear iterations with the initial guess, $\mathbf{u}_0 = \mathbf{0}$ across all the examples presented. Stopping criteria are set to

$$\|\mathbf{u}_{i+1} - \mathbf{u}_i\| < 10^{-9}$$

or the 200 maximum number of iterations.

The accuracy of the RBF-based method depends on the shape parameter if there is a shape parameter associated with the RBF. However, the determination of the optimal shape parameter is still an ad hoc topic. A statistical method called leave-one-out-cross validation (LOOCV) [10], is employed to automatically select an optimal shape parameter when MQ is used for comparison purposes.

The performance of the proposed method is tested on several nonlinear equations, including

- Example 1: a nonlinear Poisson-type equation in 2D on a square domain and several irregular domains (amoeba-like domain and peanut-shaped domain);
- Example 2: a slightly more complicated nonlinear force term when analytical solutions are simple exponential function or Franke's benchmark test function;
- Example 3: a Poisson-Boltzmann equation on a rectangular domain with Dirichlet and Neumann boundary conditions;
- Example 4: variable coefficient problem with mixed boundary conditions, and
- Example 5: a 3D nonlinear problem with bumpy sphere domain.

Example 4.1. *In this example, we consider the following nonlinear Poisson-type problem:*

$$\Delta u(x,y) = 3u^2(x,y), \qquad (x,y) \in \Omega, \tag{18}$$
$$u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega, \tag{19}$$

where $g(x,y)$ is given based on the following analytical solution

$$u(x,y) = \frac{4}{(3+x+y)^2}. \tag{20}$$

The parametric equation of the irregular domain boundary is defined as follows,

$$\partial\Omega = \{(x,y)|x = r\cos\theta, y = r\sin\theta, 0 \le \theta < 2\pi\},$$

where two irregular shapes of domains can be obtained by choosing different $r(\theta)$:

- Amoeba-like domain: $r(\theta) = 0.4e^{\sin\theta}\sin^2(2\theta) + 0.4e^{\cos\theta}\cos^2(2\theta)$
- Peanut-shaped domain: $r(\theta) = \sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}}$.

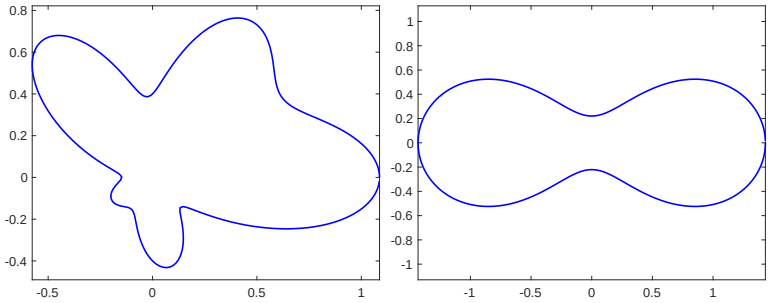

Figure 2.: Computational domains: Amoeba-shaped (left), peanut-shaped (right) used in Example 4.1.

Computational domains: Amoeba-shaped (top-left), peanut-shaped (top-right), are shown in Figure 2.

Table 2.: Example 4.1: Comparison of PHS (order 4) and MQ with additional polynomial basis of order 4 is added with $n = 35$.

Domain	(n_i, n_b)	ϵ_{rms}		ϵ_∞		
		PHS	MQ	PHS	MQ	c
Unit Square	$(841, 236)$	1.959E−09	8.764E−07	4.511E−09	3.353E−06	7.6393
Amoeba	$(861, 300)$	9.752E−10	8.218E−07	2.514E−09	7.768E−06	8.5913
Peanut	$(832, 290)$	2.788E−07	5.417E−05	9.996E−07	3.369E−04	3.1245

Table 2 shows the comparison of the performances of ILMAPS with PHS of order $k = 4$ against LMAPS with MQ when the additional polynomial basis of order $m = 4$ is added in both cases for a fair comparison. We observe high accuracy in the approximation on the unit square domain compared to results obtained on irregular domains. It is quite clear that PHS outperforms MQ regardless of the domain considered. The selection of the shape parameter of MQ is done with great care to make sure that the results of the MQ basis are almost optimal. The leave-one-out cross-validation (LOOCV) algorithm is employed with the initial search interval $[0, 10]$ to find the optimal shape parameter for the MQ basis.

Figure 3 shows a comparison of maximum absolute errors and root mean squared errors of ILMAPS on three different domains: square, amoeba, and peanut domains with different algorithm parameters:

Figure 3 Top: $k = 4, m = 6$, n various;
Figure 3 Middle: $n = 135, m = 6$, k various;
Figure 3 Bottom: $k = 6, n = 135$, m various;

As we can see, to improve the accuracy of ILMAPS, we need to employ more points in the local influence domains (up to $n = 135$) or use higher order PHS (up to $k = 12$) or even use lower degree polynomials ($m = 3$ to $m = 11$). The method is simple and easy to implement as there is no shape parameter to be selected, and small values for n, k and m are sufficient to obtain extremely accurate approximations (maximum absolute errors in the magnitude of 10^{-10}).

Example 4.2. *We consider the following nonlinear Poisson's equation with Dirichlet*

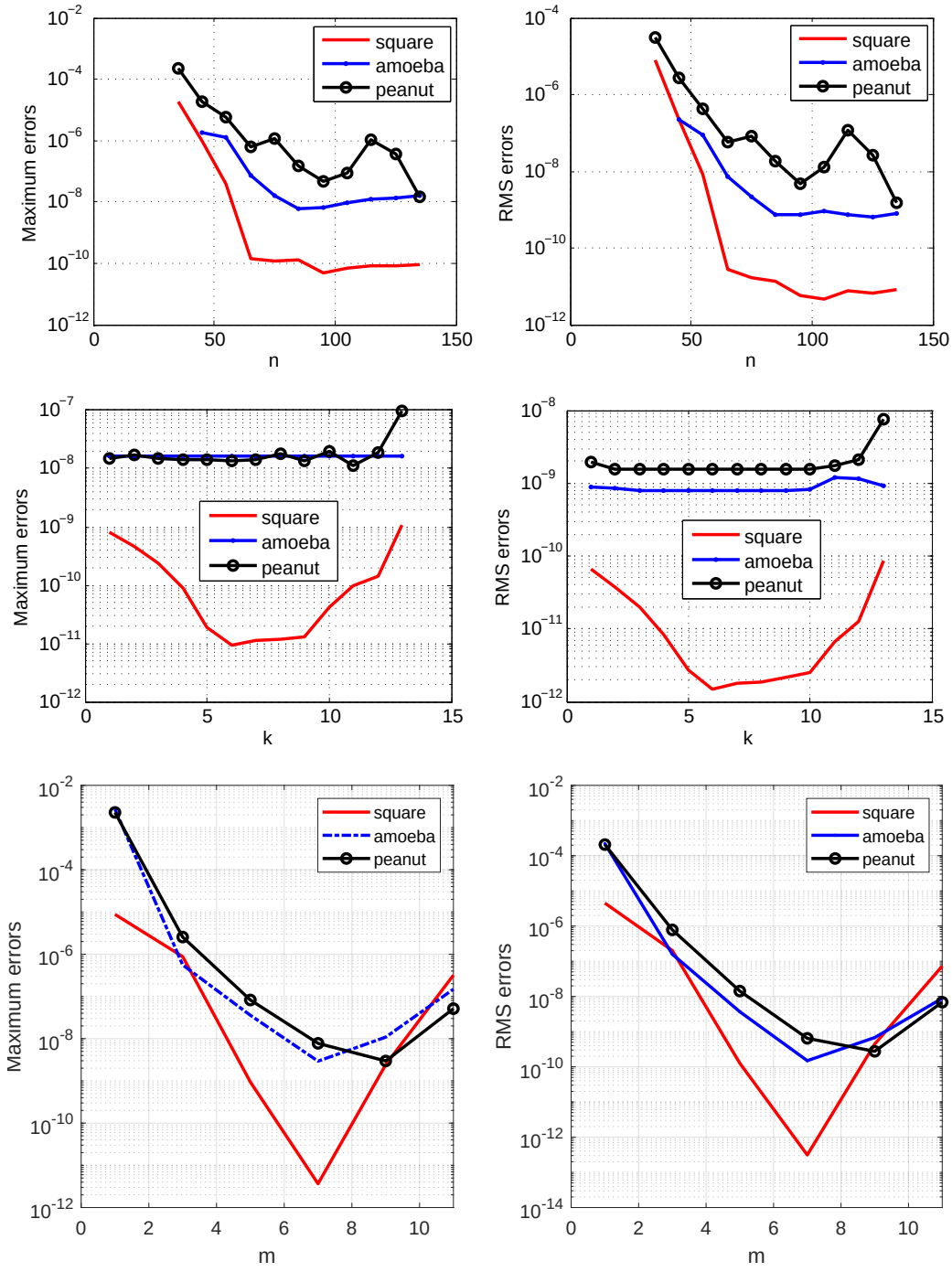

Figure 3.: Example 4.1: A comparison of maximum absolute errors and root mean squared errors of ILMAPS on three different domains: square, amoeba, and peanut domains, when the number of points in the local influence domains (n) changes (top, $k = 4, m = 6$), the order polyharmonic splines (k) changes (middle, $m = 6, n = 135$), and the order polynomials (m) changes (bottom, $k = 6, n = 135$).

boundary condition in the peanut-shaped domain as shown in Figure 2:

$$\Delta u(x,y) - 4u^3(x,y) = f(x,y) \qquad (x,y) \in \Omega, \tag{21}$$
$$u(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega, \tag{22}$$

where f and g are given based on the following analytical solutions:

- *Case 1. The analytical solution is a trivial and simple exponential function:*

$$u(x,y) = e^{x+y}, \quad (x,y) \in \overline{\Omega}. \tag{23}$$

- *Case 2. A relatively complicated analytical solution: [7]*

$$u(x,y) = (1+\sqrt{x^2+y^2})e^{-\sqrt{x^2+y^2}}, \qquad (x,y) \in \overline{\Omega}. \tag{24}$$

- *Case 3. Franke's benchmark test function [13] as the analytical solution:*

$$u(x,y) = \frac{1.25+\cos(5.4y)}{6(1+(3x-1)^2)}, \qquad (x,y) \in \overline{\Omega}. \tag{25}$$

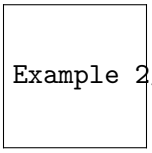

Figure 4.: Example 4.2: The profiles of the exact solutions: Case 1 is on the left, Case 2 is in the middle, and Case 3 is on the right.

Figure 5.: Example 4.2: RMS errors versus the polynomial order m for $n_i = 300$, $n_b = 100$ and $n_s = 55$ with PHS of order $k = 4$ for exact solution in (23) .

The profiles of the exact solutions are shown in Figure 4. In Table 3, we demonstrate the root mean squared error, maximum absolute errors, and the CPU time for a different number of interior and boundary nodes using PHS RBF with a polynomial basis of order 4 and 35 points in local domains of influence. It is clear that this method produces highly accurate approximations on an irregular domain, and it can handle a higher number of points in a reasonable time, regardless of the complexity of the analytical solutions. In addition, the method can achieve even higher accuracy by employing a larger number of collocation points or by increasing the size of the local domain.

It is apparent that the method can approximate the solutions accurately and efficiently despite the complexity and smoothness of the exact solution. Figure 5 shows the possibility of improving the accuracy by using a higher degree of polynomials along with a slightly higher number of local collocation points.

Table 3.: Example 4.2: ϵ_{rms}, ϵ_∞ and CPU time using different numbers of boundary and interior points with PHS of order 4, where $n = 35$ for the case of exact solution in Case 1, Case 2 and Case 3.

	(n_i, n_b)	ϵ_{rms}	ϵ_∞	CPU time (s)
Case 1	$(300, 100)$	2.197E−09	1.561E−08	3.46
	$(1488, 300)$	1.692E−09	2.172E−08	9.18
	$(6728, 600)$	3.487E−10	7.092E−09	87.11
	$(23768, 900)$	3.139E−11	5.277E−10	703.84
Case 2	$(300, 100)$	7.582E−07	3.825E−06	2.74
	$(1488, 300)$	6.023E−07	6.372E−06	8.08
	$(6728, 600)$	1.470E−07	2.191E−06	68.42
	$(23768, 900)$	3.418E−09	3.339E−08	811.19
Case 3	$(300, 100)$	1.849E−06	7.697E−06	2.24
	$(1488, 300)$	1.706E−07	2.150E−06	5.59
	$(6728, 600)$	2.170E−07	5.863E−06	45.80
	$(23768, 900)$	2.740E−09	9.450E−08	426.3

Example 4.3. *In this example, we consider the nonlinear Poisson-Boltzmann equation in a rectangular domain* $[-1, 1] \times [-1, 1]$. *The nonlinear PB equation describes many physical problems in the bio-molecular process and electrostatic interactions between colloidal particles. The governing equation is as follows:*

$$\nabla \cdot (\epsilon \nabla u(x, y)) = k^2 \sinh(u(x, y)) + f(x, y), \quad (x, y) \in \Omega, \tag{26}$$
$$u(x, y) = g(x, y), \quad (x, y) \in \partial\Omega, \tag{27}$$

where ϵ *and* k *are some known functions and for the comparison purpose, we consider the case where* $\epsilon = k = 1$,

$$f(x, y) = 4 - \sinh\left(x^2 + y^2 + e^x \cos(y)\right) \tag{28}$$
$$g(x, y) = x^2 + y^2 + e^x \cos(y). \tag{29}$$

The analytical solution is given by

$$u(x, y) = x^2 + y^2 + e^x \cos(y). \tag{30}$$

In Table 4, we present a comparison of numerical results obtained by using ILMAPS with PHS of order $k = 6$ and $m = 6$ against a method of localized form of the Moving Least Squares (MLS) [2]. The results from the ILMAPS are clearly better than what was reported in the reference for every grid configuration. It is also possible to improve the accuracy by utilizing a higher number of local points and higher degree polynomials. Figure 6 shows the contour plot of the approximated solution and the rate of convergence of ILMAPS with respect to the number of nodes N. As we can see from the figure, the rate of convergence is close to 4.141.

Next, we consider the same nonlinear Poisson-Boltzmann equation in (26) to find the distribution of electrostatic potential in a static ionic solution. Here, $k = 79, \epsilon = 1$

Table 4.: Example 4.3: Comparison of ϵ_{rms} using ILMAPS and MLS for different grid configurations with parameters $ns = 150$, $k = 6$ and $m = 6$ for ILMAPS.

Grid Size	41×41	81×81	121×121	161×161
ILMAPS	8.584E−11	6.853E−12	1.634E−13	2.316E-14
MPS [2]	7.270E−10	4.276E−11	8.281E−12	2.595E-12

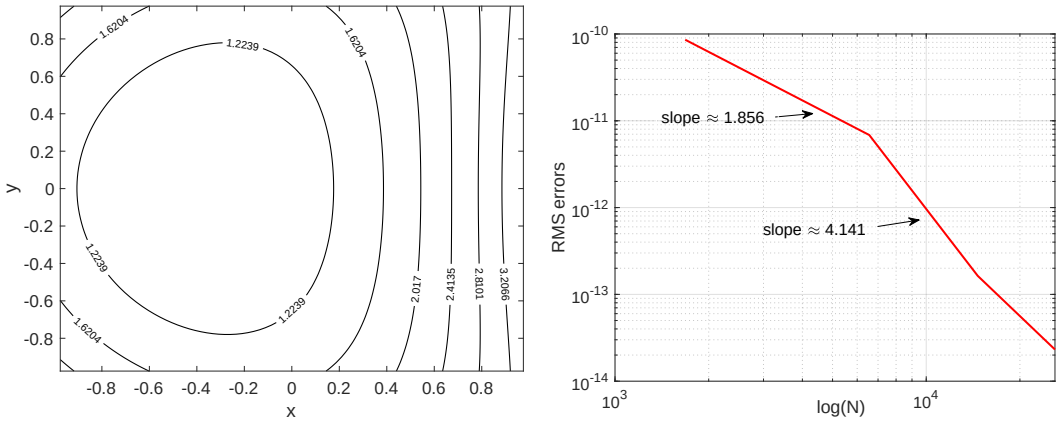

Figure 6.: Example 4.3: Contour plot of the approximate solution of part 1 on a 81 grid on the left and RMS errors versus the number of collocation points N on the right.

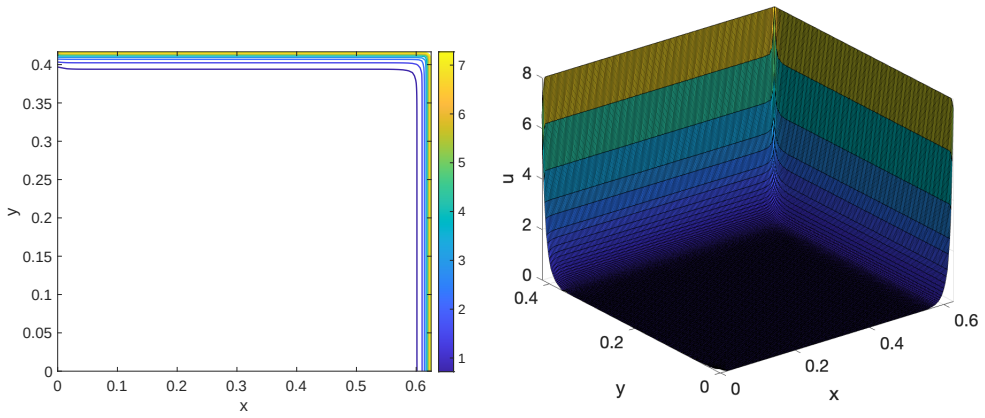

Figure 7.: Example 4.3: Contour plot of the approximate solution of part 2 on an 81×81 grid on the left and surface plot on the right.

and $f = 0$ in nonlinear Poisson-Boltzmann equation in (26) with boundary conditions

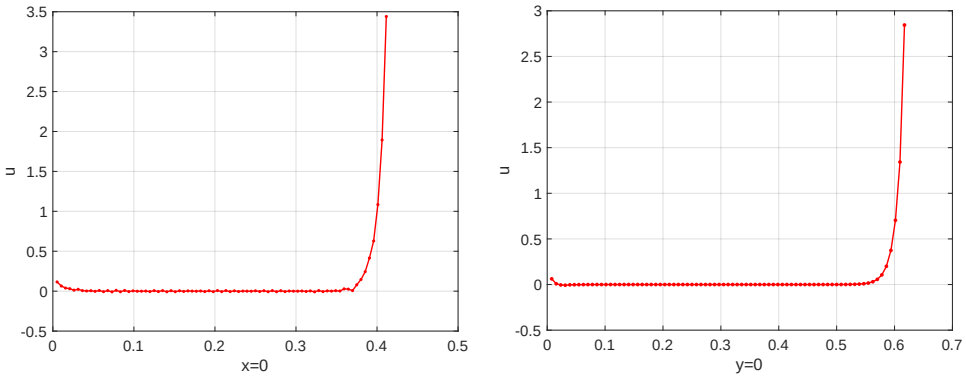

Figure 8.: Example 4.3: Profiles of the approximate solutions along $x = 0$ and $y = 0$.

defined as

$$\begin{aligned} u &= 8, && \text{if } x = 5/8, \\ u &= 8, && \text{if } y = 5/12, \\ \frac{\partial u}{\partial x} &= 0, && \text{if } x = 0, \\ \frac{\partial u}{\partial y} &= 0, && \text{if } y = 0. \end{aligned}$$

on a rectangular domain $[0, 5/8] \times [0, 5/12]$. For the comparison purpose, we use 81×81 uniformly distributed nodes in the calculations with $k = 4$, $m = 4$, and $n = 100$. Figure 7 shows the contour plot and the surface plot that is in good agreement with the corresponding figures presented in [2].

Figure 8 shows the numerical results along Neumann boundaries, $y = 0$ and $x = 0$. It is observed that the approximations along Neumann boundaries are very smooth and have no significant numerical oscillations. It can be concluded that ILMAPS is accurate, stable, and applicable to this practical problem.

Example 4.4. *In this example, we consider a nonlinear problem with variable coefficients and the mixed boundary conditions on a cassini (three) shaped domain:*

$$\begin{aligned} \Delta u(x,y) + y\cos(y)\frac{\partial u(x,y)}{\partial x} - x\sin(x)\frac{\partial u(x,y)}{\partial y} + u^2(x,y) &= f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) &= g(x,y), \quad (x,y) \in \partial\Omega^D, \\ \frac{\partial u(x,y)}{\partial \mathbf{n}} &= h(x,y), \quad (x,y) \in \partial\Omega^N, \end{aligned}$$

where $\mathbf{n}$ *is the unit outward normal vector, and* $f(x,y)$, $g(x,y)$ *and* $h(x,y)$ *are given based on the following analytical solution*

$$u(x,y) = \sin(4x)\cos(4y), \qquad (x,y) \in \overline{\Omega}. \tag{31}$$

The boundaries $\partial\Omega^D$ *and* $\partial\Omega^N$ *denote the Dirichlet and Neumann boundaries respectively such that* $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N, \partial\Omega^D \cap \partial\Omega^N = \emptyset$. *As shown on the left of Figure 9,* $\partial\Omega^N$ *is defined in the fourth quadrant; i.e.,* $3\pi/2 \leq \theta < 2\pi$.

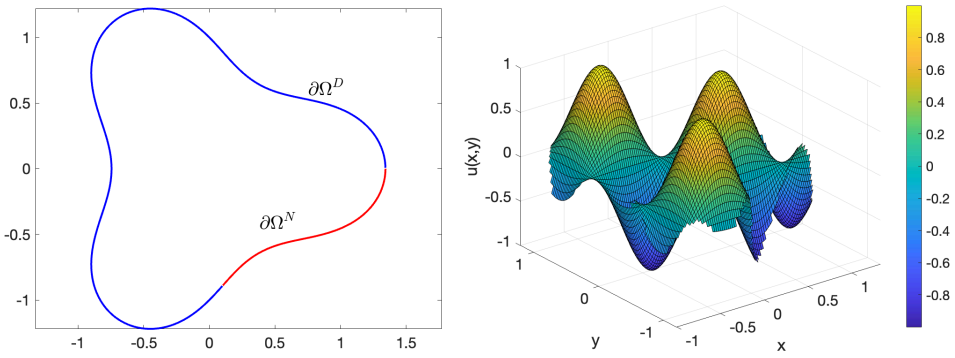

Figure 9.: Example 4.4: The profiles of the computational domain and the analytical solution.

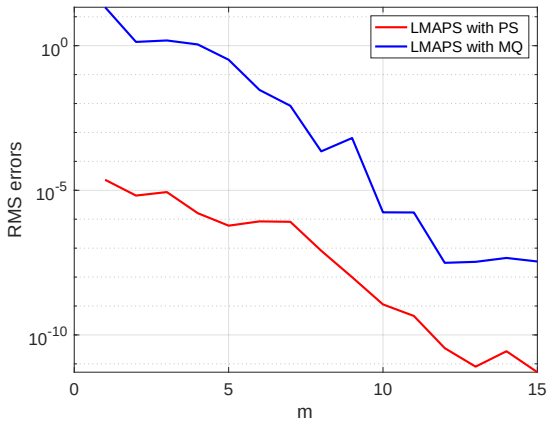

Figure 10.: Example 4.4: RMS errors versus the polynomial order m with $n_i = 6830$, $n_b = 400$, $n = 150$ and for LMAPS with PHS $k = 4$.

The cassini (three)-shaped domain boundary is defined by the following parametric equation:

$$\partial\Omega = \{(x, y)|x = r\cos\theta, y = r\sin\theta,\ \ 0 \leq \theta < 2\pi\}$$

where $r(\theta) = \left(\cos(3\theta) + \sqrt{2 - \sin^2(3\theta)}\right)^{1/3}$.

Table 5 shows the numerical results obtained using various orders of polynomial basis. For the sake of comparison with the results reported in [5] for LMAPS with polynomial basis functions (PBF), we choose 14350 interior nodes, 307 nodes on the Dirichlet boundary, and 93 nodes on the Neumann boundary. The order of PHS is 4 and 55 local nodes are used to obtain the results for ILMAPS. We see that accuracy increases as the order of polynomials gets higher and ILMAPS is always a few orders of magnitude better than LMAPS with PBF. However, LMAPS with PBF performs better than ILMAPS in terms of efficiency. However, the same order of accuracy and level of efficiency as reported in the reference can be achieved with a smaller number

Table 5.: Example 4.4: Comparison of ϵ_{rms}, ϵ_∞ using ILMAPS with PHS and LMAPS with PBF for different order of polynomial basis with $n_i = 14,350$, $n_b = 400$.

m	LMAPS with PHS		LMAPS with PBF [5]	
	ϵ_{rms}	ϵ_∞	ϵ_{rms}	ϵ_∞
3	9.915E−07	2.406E−06	2.96E−03	2.73E−02
4	8.324E−08	2.785E−07	5.34E−04	2.91E−03
5	6.426E−08	1.670E−07	7.70E−06	6.19E−05
6	2.605E−09	3.924E−08	5.77E−06	5.53E−05

of collocation points. As an example, it only takes 1703 interior nodes with the same number of boundary points to achieve accuracy up to the order of 10^{-8} with lesser CPU time than it is for LMAPS.

In Figure 10 we compare ILMAPS against LMAPS with MQ RBF using $n_i = 6830, n_b = 400, n = 150$ for MQ RBF and same set of parameter values and additionally $k = 4$ for PHS. It is clear that as we further increase the highest order of polynomial basis up to the order of 15, the numerical results get extremely accurate and are also better than the results from MQ RBF. Further increase of m causes instability in numerical results and even loss of accuracy as well as an increase of computational time.

Example 4.5. *In this example, we consider the following 3D problem on a bumpy sphere domain:*

$$\Delta u(x,y,z) = \frac{2}{u(x,y,z)} + \frac{3}{u^3(x,y,z)}, \quad (x,y,z) \in \Omega, \tag{32}$$

$$u(x,y,z) = g(x,y,z), \quad (x,y,z) \in \partial\Omega, \tag{33}$$

where $g(x,y,z)$ is given based on the following analytical solution

$$u(x,y,z) = \sqrt{3 + x^2 + y^2 + z^2}, \qquad (x,y,z) \in \overline{\Omega}. \tag{34}$$

The boundary of the bumpy sphere domain is defined as follows:

$$\partial\Omega = \{(x,y,z) | x = r\cos(\vartheta)\cos(\theta), y = r\cos(\vartheta)\sin(\theta), z = r\sin(\vartheta), \vartheta \in [0,\pi], \theta \in [0,2\pi]\},$$

where $r = 1 + \frac{1}{6}\sin(6\theta)\sin(7\vartheta)$.

To demonstrate the effectiveness of the method for higher dimensions, we consider this example on a quite complicated 3D domain as shown in Figure 11 (left). Table 6 shows the maximum absolute errors and the root mean squared errors using ILMAPS with PHS of order 2 and MQ RBF for various orders of polynomial basis m, where $n_i = 8830$, $n_b = 700$ and $n = 100$. Figure 11 (right) shows the exact solution on the surface of the bumpy sphere domain where the color of the surface represents the analytical solution's values at that location. As seen in the 2D examples, ILMAPS with PHS performs better than MQ basis and the accuracy increases as the degree of the polynomial basis gets higher. The shape parameter of MQ has to be determined carefully for a fair comparison with the proposed method. This is done using LOOCV with the initial search interval $[0, 5]$. As there is no need to determine a shape parameter

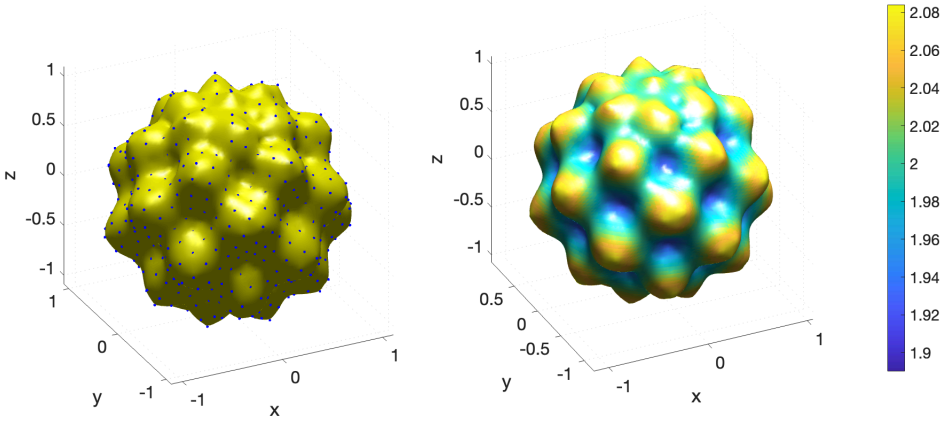

Figure 11.: Example 4.5: The profile of the bumpy sphere with boundary points on the left and the profile of the exact solution on the boundary surface on the right.

for PHS, solving 3D problems is as simple as that for 2D problems.

Table 6.: Example 4.5: ϵ_{rms} and ϵ_∞ using different orders of polynomial basis m with PHS of order 2 and MQ.

m	ϵ_{rms}		ϵ_∞		
	PHS	MQ	PHS	MQ	c
2	2.009E−05	1.136E−05	9.197E−04	7.802E−05	3.175
3	1.147E−05	1.130E−05	3.969E−05	9.164E−05	3.175
4	6.889E−07	5.643E−06	3.021E−05	3.189E−05	3.175
5	7.472E−08	2.141E−06	2.044E−06	3.129E−05	3.175

5. Conclusion and future work

In this paper, the applicability of the improved localized method of particular solutions (ILMAPS) using polyharmonic splines (PHS) with a polynomial basis for solving nonlinear elliptic PDEs in two- and three-dimensional spaces has been demonstrated. We found that

- The performance of the method is examined on five different examples on regular and irregular domains with Dirichlet and Neumann boundary conditions.
- The method is not only robust and easy to implement but the accuracy of the numerical results is high.
- Comparisons of the numerical results with results obtained using ILMAPS with MQ basis, LMAPS with polynomial basis functions (PBF), and method of localized form of moving least squares (MLS) show that ILMAPS are among the most accurate for solving nonlinear PDEs.
    - Numerical simulations clearly show that the accuracy and the efficiency of ILMAPS are superior to that of LMAPS with MQ RBF and it also resolves the issue of searching for an appropriate value of the shape parameter in MQ.
    - In addition, the numerical results were found to be more accurate than

LMAPS with PBF and it is competitive with the method of MLS as well.

- Our numerical experiments revealed that using more local points and higher order polynomial basis will improve the accuracy of ILMAPS up to a certain level.
- ILMAPS can perform the same level of efficiency as other methods if the number of local points and the highest order of polynomial basis are chosen appropriately, especially for large-scale problems.

It is worth mentioning the differences between the original LMAPS and the Improved LMAPS and why ILAMPS might replace the original LMAPS. ILMAPS uses polyharmonic splines and a low degree of polynomial basis to approximate solutions whereas original LMAPS is limited to Gaussian, MQ, or Matern RBFs. Advantages of improved LMAPS are, that the difficulty of searching for the optimal shape parameter is alleviated, accuracy is improved simply by increasing the order of polyharmonic splines or polynomial basis, or the number of interpolation points, and better stability with the use of additional polynomial basis.

In our future work, we hope to implement the method for problems with higher-order differential operators and other types of nonlinear problems. We have shown that the method is very accurate, we will also want to investigate the strategies to improve the computational efficiency without a loss of the accuracy of the method. Picard's method was used for the nonlinear iterations in our simulations and the use of other nonlinear solvers such as Newton's method will be subject to further studies as well.

Acknowledgment:

This research was supported in part by funding from the Simons Foundation and the Centre de Recherches Mathématiques, through the Simons-CRM scholar-in-residence program. The fourth author was supported by an NSERC Discovery Grant, and by the fellowship grant P2021-4201 of the National University of Mongolia.

References

[1] P.S. Bernard, *A deterministic vortex sheet method for boundary layer flow*, Journal of Computational Physics 117 (1995), pp. 132–145.

[2] G. Bourantas and V. Burganos, *An implicit meshless scheme for the solution of transient non-linear poisson-type equations*, Engineering Analysis with Boundary Elements 37 (2013), pp. 1117–1126.

[3] C. Chen, C.M. Fan, and P. Wen, *The method of approximate particular solutions for solving elliptic problems with variable coefficients*, International Journal of Computational Methods 8 (2011), pp. 545–559.

[4] G. Dahlquist and Å. Björck, *Numerical methods in scientific computing, volume i*, SIAM, 2008.

[5] T. Dangal, B. Khatri Ghimire, and A. Lamichhane, *Localized method of particular solutions using polynomial basis functions for solving two-dimensional nonlinear partial differential equations*, Partial Differential Equations in Applied Mathematics 4 (2021), p. 100114.

[6] T. Dangal, C. Chen, and J. Lin, *Polynomial particular solutions for solving elliptic partial differential equations*, Computers & Mathematics with Applications 73 (2017), pp. 60–70.

[7] F. Dou, Y. Liu, and C. Chen, *The method of particular solutions for solving nonlinear poisson problems*, Computers & Mathematics with Applications 77 (2019), pp. 501–513.
[8] R. Eymard, T. Gallouët, and R. Herbin, *Finite volume methods*, Handbook of numerical analysis 7 (2000), pp. 713–1018.
[9] C.M. Fan, C.H. Yang, and W.S. Lai, *Numerical solutions of two-dimensional flow fields by using the localized method of approximate particular solutions*, Engineering Analysis with Boundary Elements 57 (2015), pp. 47–57.
[10] G.E. Fasshauer and J.G. Zhang, *On choosing "optimal" shape parameters for rbf approximation*, Numerical Algorithms 45 (2007), pp. 345–368.
[11] G.F. Fasshauer, *Meshfree Approximation Methods with MATLAB*, World Scientific Publishing Co., Inc., USA, 2007.
[12] W. Florez, M. Portapila, N. Caruso, D. Castro, C. Bustamante, R. Posada, and J. Granados, *Numerical examination of the effect of different boundary conditions on the method of approximate particular solutions for scalar and vector problems*, Engineering Analysis with Boundary Elements 127 (2021), pp. 75–90.
[13] R. Franke, *Scattered data interpolation: Tests of some method*, Mathematics of Computation 38 (1982), pp. 181–200.
[14] M.A. Golberg, *The method of fundamental solutions for poisson's equation*, Engineering Analysis with Boundary Elements 16 (1995), pp. 205–213.
[15] E.J. Kansa, *Multiquadrics: A scattered data approximation scheme with applications to computational fluid-dynamics ii solutions to parabolic, hyperbolic and elliptic partial differential equations*, Computers & mathematics with applications 19 (1990), pp. 147–161.
[16] J. Li, Y. Hon, and C. Chen, *Numerical comparisons of two meshless methods using radial basis functions*, Eng. Anal. Boundary Elements 26 (2002), pp. 205–225.
[17] M. Li, G. Amazzar, A. Naji, and C. Chen, *Solving biharmonic equation using the localized method of approximate particular solutions*, International Journal of Computer Mathematics 91 (2014), pp. 1790–1801.
[18] W. Li, G. Yao, and J. Niu, *The modified localized method of approximated particular solutions for solving elliptic equations with mixed boundary conditions on scattered data*, Engineering Analysis with Boundary Elements 100 (2019), pp. 164–174.
[19] C. Lin, M. Gu, D. Young, and C. Chen, *Localized method of approximate particular solutions with cole–hopf transformation for multi-dimensional burgers equations*, Engineering Analysis with Boundary Elements 40 (2014), pp. 78–92.
[20] J. Lin, Y. Hong, L.H. Kuo, and C.S. Liu, *Numerical simulation of 3d nonlinear schrödinger equations by using the localized method of approximate particular solutions*, Engineering Analysis with Boundary Elements 78 (2017), pp. 20–25.
[21] T. Liszka and J. Orkisz, *The finite difference method at arbitrary irregular grids and its application in applied mechanics*, Computers & Structures 11 (1980), pp. 83–95.
[22] G.R. Liu, *Meshfree methods: moving beyond the finite element method*, CRC press, 2009.
[23] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, Vol. 113, Springer, 2016.
[24] T. Rabczuk and T. Belytschko, *Cracking particles: a simplified meshfree method for arbitrary evolving cracks*, International journal for numerical methods in engineering 61 (2004), pp. 2316–2343.
[25] T. Rabczuk and T. Belytschko, *A three-dimensional large deformation meshfree*

method for arbitrary evolving cracks, Computer methods in applied mechanics and engineering 196 (2007), pp. 2777–2799.

[26] S.A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics*, Computers & structures 83 (2005), pp. 1526–1535.

[27] L. Su, *Numerical solution of two-dimensional nonlinear sine-gordon equation using localized method of approximate particular solutions*, Engineering Analysis with Boundary Elements 108 (2019), pp. 95–107.

[28] F. Ureña, E. Salete, J.J. Benito, and L. Gavete, *Solving third- and fourth-order partial differential equations using gfdm: application to solve problems of plates*, International Journal of Computer Mathematics 89 (2012), pp. 366–376.

[29] L. Wu and Y.K. Kwok, *A front-fixing finite difference method for the valuation of american options*, Journal of Financial Engineering 6 (1997), pp. 83–97.

[30] G. Yao, *Local radial basis function methods for solving partial differential equations*, PhD Dissertation, University of Southern Mississippi, 2010.

[31] G. Yao, *An improved localized method of approximate particular solutions for solving elliptic pdes*, Computers & Mathematics with Applications 71 (2016), pp. 171–184.

[32] G. Yao, C.S. Chen, and H. Zheng, *A modified method of approximate particular solutions for solving linear and nonlinear pdes: Modified method of approximate particular solutions*, Numerical Methods for Partial Differential Equations 33 (2017).

[33] G. Yao, C. Chen, W. Li, and D. Young, *The localized method of approximated particular solutions for near-singular two-and three-dimensional problems*, Computers & Mathematics with Applications 70 (2015), pp. 2883–2894.

[34] G. Yao, J. Kolibal, and C.S. Chen, *A localized approach for the method of approximate particular solutions*, Computers & Mathematics with Applications 61 (2011), pp. 2376–2387.

[35] Y. You, J.S. Chen, and H. Lu, *Filters, reproducing kernel, and adaptive meshfree method*, Computational mechanics 31 (2003), pp. 316–326.

[36] X. Zhang, M. Chen, C. Chen, and Z. Li, *Localized method of approximate particular solutions for solving unsteady navier–stokes problem*, Applied Mathematical Modelling 40 (2016), pp. 2265–2273.