



---

*Research article*

## Smooth digital terrain modeling in irregular domains using finite element thin plate splines and adaptive refinement

Lishan Fang\*

School of Mathematical Sciences, Huaqiao University, Quanzhou 362021, China

\* **Correspondence:** Email: [fanglishan@hqu.edu.cn](mailto:fanglishan@hqu.edu.cn).

**Abstract:** Digital terrain models (DTMs) are created using elevation data collected in geological surveys using varied sampling techniques like airborne lidar and depth soundings. This often leads to large data sets with different distribution patterns, which may require smooth data approximations in irregular domains with complex boundaries. The thin plate spline (TPS) interpolates scattered data and produces visually pleasing surfaces, but it is too computationally expensive for large data sizes. The finite element thin plate spline (TPSFEM) possesses smoothing properties similar to those of the TPS and interpolates large data sets efficiently. This article investigated the performance of the TPSFEM and adaptive mesh refinement in irregular domains. Boundary conditions are critical for the accuracy of the solution in domains with arbitrary-shaped boundaries and were approximated using the TPS with a subset of sampled points. Numerical experiments were conducted on aerial, terrestrial, and bathymetric surveys. It was shown that the TPSFEM works well in square and irregular domains for modeling terrain surfaces and adaptive refinement significantly improves its efficiency. A comparison of the TPSFEM, TPS, and compactly supported radial basis functions indicates its competitiveness in terms of accuracy and cost.

**Keywords:** smoothing; thin plate spline; mixed finite element; adaptive mesh refinement; digital terrain model

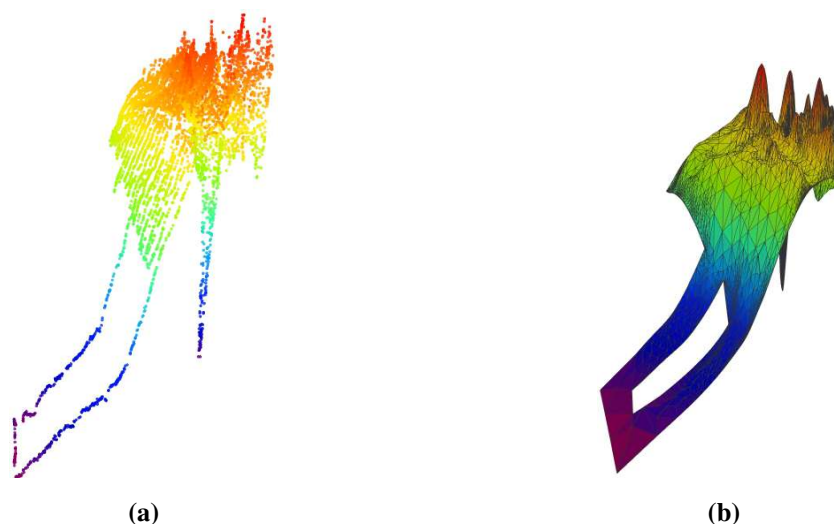
**Mathematics Subject Classification:** 65D10, 65N30, 68U05

---

### 1. Introduction

Digital terrain models (DTMs) represent continuous terrain surfaces constructed using scattered data from geological surveys. They have been widely applied in various areas, including geological modeling, mining engineering, and geomorphological mapping [16, 19]. The DTMs are normally built by interpolating observed data of terrain surfaces. An example is shown in Figure 1, where irregularly distributed data are interpolated in an irregularly shaped domain with an interior hole.

Popular interpolation techniques for the DTMs include wavelet-based methods [1], kriging [30], inverse distance weighting, and smoothing splines [6], whose performance is influenced by the sizes and distribution patterns of data. Wavelet smoothing methods are most useful for data observed on a regular mesh in a rectangular domain and radial basis functions (RBFs) are applicable for arbitrarily scattered data [24]. However, computational costs of RBFs like the thin plate spline (TPS) become impractical for large data sets. Methods like the random feature-based method [7] and triangulated irregular network [21] were proposed to achieve higher efficiency, but they may not possess the flexibility and smoothing properties of the RBFs.



**Figure 1.** (a) Observed data of a bathymetric survey; and (b) the corresponding interpolant.

Many techniques were developed to improve the efficiency of the RBFs. Some use basis functions with compact support instead of global support for sparser systems of equations. Wendland [29] presented several RBFs with compact support and DeParis et al. [9] introduced a scheme to adjust the radius of support for irregularly distributed data. Some others, like the residual subsampling method, produce smaller systems of equations [11]. Another approach is to develop fast solvers for RBF interpolations. Beatson et al. [2] used fast matrix-vector multiplication and a generalized minimal residual iteration with pre-conditioners to reduce the costs for calculating RBFs.

Smooth interpolations based on second- or higher-order finite elements (FEs) have also been developed for geological modeling. The finite element method (FEM) is often used to model complex models and works well in arbitrary topography. Briggs [3] did some early work and proposed a discretized minimum curvature interpolation of the TPS to draw contour maps for the geophysical data. Ramsay [24] introduced a discrete bivariate smoother based on quadratic FEs for smoothing over domains with complex irregular boundaries or holes. Recently, Chen et al. [8] approximated the TPS using nonconforming Morley FEs and provided stochastic convergence analysis.

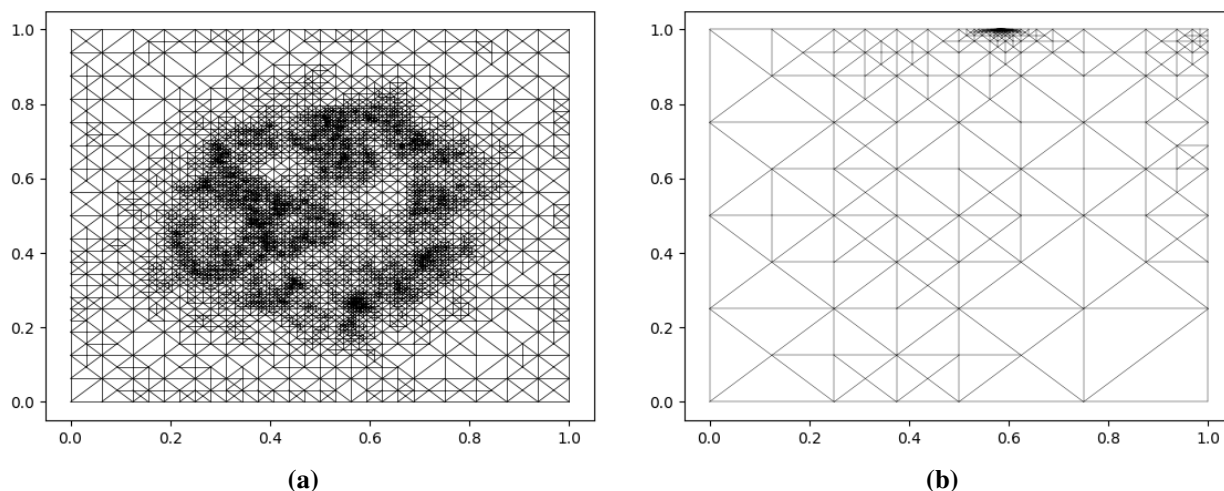
The finite element thin plate spline (TPSFEM) was developed by Roberts et al. [25] to interpolate and smooth large data sets of surface fitting and data mining applications. It is the first to use first-order FEs for a sparser system of linear equations that achieves higher efficiency [28]. Adaptive mesh refinement and four error indicators were adapted from partial differential equations (PDEs) for the TPSFEM by Fang and Stals [14], which further enhanced its efficiency. Experiments showed that

adaptive TPSFEM achieved lower root mean square error (RMSE) using less than 40% of nodes and 50% of system-solving time in square and L-shaped domains compared to uniform FE meshes.

Irregular domains with complex boundaries are often required in applications like computer graphics, geometric modeling and geospatial modeling [4, 18]. They better reflect the geometry of the domain of interest and more accurately model surfaces compared to uniform square domains. However, irregular domains pose challenges to traditional RBFs, which rely on the Euclidean distance to measure the closeness between two points. While the RBFs work in domains with convex boundaries and no holes, they may smooth across concavities in the boundary or around the holes for irregular domains [24]. An example is shown in Figure 2, where lines connecting points A and B fall outside the domain. While these two points are close in Euclidean distance, they may not have a significant correlation.



**Figure 2.** (a) U-shaped domain; and (b) square domain with an interior hole.



**Figure 3.** Adaptively refined mesh (a) without over-refinement; and (b) with over-refinement near  $[0.6, 1.0]$ ; for the same data.

This problem can be addressed using FEs to define the irregular topography. Ramsay's bivariate smoother interpolates on irregular domains but is expensive for large data sets [24]. The TPSFEM interpolates large data sets efficiently and adaptive refinement markedly improves its efficiency in both square and L-shaped domains [13, 14]. However, the TPSFEM has two major problems with smoothing irregularly scattered data in irregularly shaped domains, which Ramsay stated is the most difficult practical scenario. First, boundary conditions are critical for the accuracy of the TPSFEM and adaptive refinement but may be unknown for real-world data sets. Fang and Stals set Dirichlet boundaries as

a constant number close to values of data, which will not work for irregular domains with varied data values close to boundaries. Accurate and efficient approximations of boundary conditions are required to build the TPSFEM in irregular domains. More information is given in Section 4. Second, over-refinement of FEs on oscillatory surfaces was found near the boundaries of FE meshes. Figure 3(a) shows an adaptive mesh that captures the oscillatory surfaces of a geological survey, which improves the efficiency of the resulting TPSFEM. In contrast, the mesh in Figure 3(b) for the same data has over-refinement of FEs near  $[0.6, 1.0]$  while the other FEs remain coarse. Consequently, the efficiency of the resulting TPSFEM diminishes. This behavior may become more severe in irregular domains with arbitrary-shaped boundaries and requires further investigation.

This article focuses on applications of the TPSFEM and adaptive mesh refinement for the DTMs in irregular domains. Dirichlet boundary conditions are approximated using the TPS with a small subset of sampled points. Over-refinement observed in previous studies becomes prevalent near complex boundaries in irregular domains and is alleviated by adjusting refinement routines. Numerical experiments are conducted on three real-world geological surveys to evaluate the performance of the TPSFEM and adaptive refinement in square and irregular domains. The analysis of the results focuses on the impacts of irregular domains and data distribution patterns on the TPSFEM and adaptive refinement. Finally, a comparison of the TPSFEM to the TPS and two compactly supported RBFs is given to validate the TPSFEM's effectiveness.

The remainder of this article is organized as follows. Square and irregular domains are compared in Section 2. The formulation and adaptive refinement of the TPSFEM is provided in Section 3. The Dirichlet boundary conditions approximated using the TPS are discussed in Section 4. The numerical experiments and data sets are described in Section 5. Results and discussions of the experiments are given in Section 6. Conclusions of findings are summarized in Section 7.

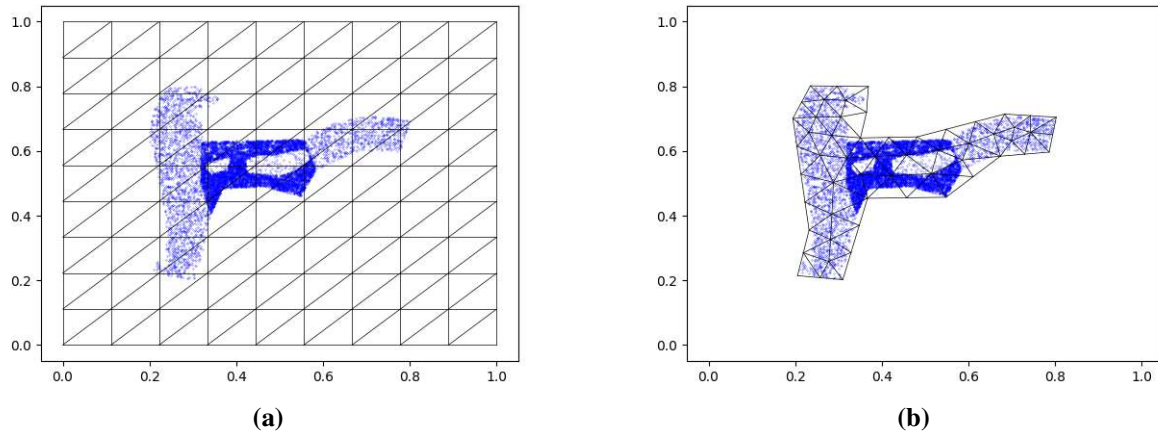
## 2. Square and irregular domains

Recall that we are working with a data-fitting technique based on FE discretization. The first step of building the TPSFEM is constructing an FE mesh to contain the domain of the observed data. This article focuses on adaptive refinement over two-dimensional FE meshes with triangular elements and piecewise linear basis functions. The domain is partitioned into disjoint triangular elements and data points will be covered by at least one triangle. Examples of the observed data and corresponding FE meshes in square or irregular domains are shown in Figure 4.

Square domains have been used to analyze properties of the TPSFEM and are applicable for data with various distribution patterns as shown in Figure 4(a). However, they may lack accurate representations of the exterior shape of interpolants and cannot be used for modeling objects in complex geometric forms. Irregular domains fit the data closely and better reflect the domain of interest as shown in Figure 4(b). The shape of irregular domains depends heavily on the configuration of data points, which often vary across applications. Aerial surveys collect data through flights that form frames and are less affected by terrain. In contrast, ground or bathymetric surveys may be limited by obstacles like the bank of a water body and sampled data have complex external geometric shapes. The construction of FE meshes in irregular domains is described in Supplementary A.

A potential benefit of irregular domains is the higher efficiency compared to square domains. Many elements in the FE mesh in Figure 4(a) do not contain any data points and may contribute little to

improving the accuracy of the solution. In contrast, most elements in Figure 4(b) cover data points and the resulting interpolant may have higher efficiency. Note that this characteristic may overlap with adaptive mesh refinement and the impacts of their combinations are explored in Section 6.



**Figure 4.** FE meshes built in (a) a square domain; and (b) an irregular domain; with triangular elements. The data points are represented as blue dots.

### 3. Finite element thin plate spline

#### 3.1. Formulation

Important concepts and notions of the TPS and TPSFEM are introduced in this section for further discussion. They have been explained in more detail in [25]. The interpolation  $t$  of traditional TPS minimizes functional

$$J_{\alpha}(t) = \frac{1}{n} \sum_{i=1}^n (t(\mathbf{x}_i) - y_i)^2 + \alpha \int_{\Omega} \sum_{|\mathbf{v}|=2} \binom{2}{\mathbf{v}} (D^{\mathbf{v}}t(\mathbf{x}))^2 d\mathbf{x} \quad (3.1)$$

over  $H^2(\Omega)$ , where  $\mathbf{v} = (v_1, \dots, v_d)$ ,  $|\mathbf{v}| = \sum_{j=1}^d v_j$ , and  $v_j \in \mathbb{N}$  for  $j = 1, \dots, d$ . Smoothing parameter  $\alpha$  balances the conformity to the data and smoothness of  $t$ . The construction of the TPS and an evaluation of its interpolant at a given point require  $O(n^3)$  and  $O(n)$  arithmetic operations, respectively. As  $n$  increases, the TPS becomes more accurate but computationally expensive.

The TPSFEM is defined on an FE mesh  $\mathcal{T}$  of dimension  $d$ , which covers the domain of interest  $\Omega$ . Given observed data  $\{(\mathbf{x}_i, y_i) : i = 1, 2, \dots, n\}$  of size  $n$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  is the  $i$ -th  $d$ -dimensional predictor value and  $y_i \in \mathbb{R}$  is the  $i$ -th response value. The TPSFEM  $s(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}$  is represented as a combination of  $m$  piecewise linear basis functions  $\mathbf{b}$ , where  $\mathbf{b} = [b_1, \dots, b_m]^T$  and  $\mathbf{c}$  are corresponding coefficients. The basis function  $b_k$  corresponds to the  $k$ -th node and  $b_k(\mathbf{x}) = 0$  if  $\mathbf{x}$  lies outside of the elements that contain the  $k$ -th node.

The minimization problem of the TPSFEM was adapted from Eq (3.1), which requires continuous high-order derivatives to measure the smoothness of interpolants. While first-order derivatives  $\nabla s$  can be computed using  $s$  directly, the approximation of its second-order derivatives will be inaccurate [33]. Roberts et al. [25] introduced auxiliary functions  $\mathbf{u} = [u_1, \dots, u_d]^T$  to represent the gradient of  $s$ , where  $u_j$  approximate the gradients of  $s$  in dimension  $j$  for  $1 \leq j \leq d$ . Function  $u_j$  is defined as  $u_j(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{g}_j$  and  $\mathbf{g}_j$  are the corresponding coefficients.

The TPSFEM  $s$  minimizes

$$\begin{aligned} J(\mathbf{c}, \mathbf{g}_1, \dots, \mathbf{g}_d) &= \frac{1}{n} \sum_{i=1}^n (\mathbf{b}(\mathbf{x}_i)^T \mathbf{c} - y_i)^2 + \alpha \int_{\Omega} \sum_{j=1}^d \nabla (\mathbf{b}(\mathbf{x})^T \mathbf{g}_j) \nabla (\mathbf{b}(\mathbf{x})^T \mathbf{g}_j) \, d\mathbf{x} \\ &= \mathbf{c}^T A \mathbf{c} - 2\mathbf{d}^T \mathbf{c} + \mathbf{y}^T \mathbf{y} / n + \alpha \sum_{j=1}^d \mathbf{g}_j^T L \mathbf{g}_j, \end{aligned} \quad (3.2)$$

subject to

$$L\mathbf{c} = \sum_{j=1}^d G_j \mathbf{g}_j, \quad (3.3)$$

where  $\mathbf{d} = \frac{1}{n} \sum_{i=1}^n \mathbf{b}(\mathbf{x}_i) y_i$ ,  $\mathbf{y} = [y_1, \dots, y_n]^T$ , and  $A = \frac{1}{n} \sum_{i=1}^n \mathbf{b}(\mathbf{x}_i) \mathbf{b}(\mathbf{x}_i)^T$ . Matrix  $L$  is a discrete approximation to the negative Laplacian and  $L_{pq} = \int_{\Omega} \nabla \mathbf{b}_p \nabla \mathbf{b}_q \, d\mathbf{x}$ . Matrix  $G_j$  is a discrete approximation to the gradient operator in dimension  $j$  for  $1 \leq j \leq d$  and  $(G_j)_{pq} = \int_{\Omega} \mathbf{b}_p \partial_j \mathbf{b}_q \, d\mathbf{x}$ . The information of the data is projected onto the FE mesh and stored in matrix  $A$  and vector  $\mathbf{d}$ . Equation (3.3) ensures that  $\nabla s$  and  $\mathbf{u}$  are equivalent in a weak sense. Smoothing parameter  $\alpha$  is calculated using generalized cross-validation (GCV) [28].

The minimization problem in Eq (3.2) subject to Eq (3.3) is rewritten as a system of linear equations using Lagrange multipliers. For example, the system in a two-dimensional domain is written as

$$\begin{bmatrix} A & \mathbf{0} & \mathbf{0} & L \\ \mathbf{0} & \alpha L & \mathbf{0} & -G_1^T \\ \mathbf{0} & \mathbf{0} & \alpha L & -G_2^T \\ L & -G_1 & -G_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \end{bmatrix},$$

where  $\mathbf{w}$  is a Lagrange multiplier and  $\mathbf{h}_1$ ,  $\mathbf{h}_2$ ,  $\mathbf{h}_3$ , and  $\mathbf{h}_4$  store Dirichlet boundary information. The system's dimension is independent of the number of data points  $n$  instead of the number of basis functions  $m$ , which improves its scalability. This system is ill-conditioned for small  $h$  and small  $\alpha$  as with the system of the TPS from Eq (3.1), which may affect its applicability. Stals [27] developed a preconditioned conjugate gradient solver for the TPSFEM, which is well-conditioned for a wide range of  $\alpha$  and  $h$ .

The error convergence of the TPSFEM smoother  $s$  is dependent on  $\alpha$ , mesh size  $h$ , and maximum distance between data points  $d_X$  [14]. While refinement of FEs reduces  $h$ , it cannot change  $d_X$  and the RMSE of  $s$  may not be reduced. For example, adaptively refining a small region near the boundaries may lead to small  $h$  with relatively large  $d_X$ , where the RMSE of  $s$  is not reduced. This is undesired as it increases computational costs and the efficiency diminishes.

### 3.2. Iterative adaptive refinement process

The efficiency of the TPSFEM may be improved by adaptive refinement that iteratively refines elements in sensitive regions like peaks and jumps marked by error indicators. The resulting adaptively refined meshes achieve higher precision with fewer nodes than uniform meshes, which is critical for smoothing large data sets. Detailed descriptions of the error indicators and iterative adaptive refinement process are given in [14] and a brief version is provided in Supplementary B.

Four error indicators were adapted from FE approximations for PDEs in [14] and we concentrate on the auxiliary problem and recovery-based error indicators. The auxiliary problem error indicator approximates the error of the solution using the difference between interpolant  $s$  and a locally more accurate approximation. The recovery-based error indicator post-processes gradients of  $s$  to estimate errors. The former accesses neighboring data points directly and the latter only uses values in FE meshes.

An iterative adaptive refinement process was adapted for the TPSFEM as described in Algorithm 1, where  $\mathcal{T}_k$ ,  $\alpha_k$ , and  $s_k$  represent the FE mesh, smoothing parameter, and TPSFEM smoother in the  $k$ -th iteration, respectively. This algorithm does not depend on a preset error tolerance value and employs an inner loop from steps 5 to 8 to iteratively refine FEs. The inner loop marks and refines 10% of base edges with the highest error indicator values. After error indicator values of new base edges are calculated, this process restarts until the number of nodes doubles. Our approach is similar to the sorting strategy described in [23], except that we do not solve the system inside the inner loop for higher efficiency. Some studies like [12, 32] optimized the effectiveness of this process using deep learning, which may be considered for further improving this technique.

---

**Algorithm 1** Iterative adaptive refinement process of TPSFEM

---

**Input:** Data and initial FEM grid  $\mathcal{T}_0$

**Output:** TPSFEM smoother  $s_k$

- 1: Build initial mesh  $\mathcal{T}_0$ , calculate optimal  $\alpha_0$ , and build TPSFEM  $s_0$  on  $\mathcal{T}_0$ ; calculate RMSE of  $s_0$ .
  - 2:  $k \leftarrow 1$ .
  - 3: **while** RMSE is higher than the error tolerance or is not reduced less than 10% for two iterations **do**
  - 4:     calculate error indicators for the edges in mesh  $\mathcal{T}_k$ .
  - 5:     **while** the size of  $\mathcal{T}_k$  is smaller than twice the size of  $\mathcal{T}_{k-1}$  **do**
  - 6:         Iteratively mark and refine a subset of elements with high error indicator values.
  - 7:         Calculate error indicator values for the new edges in  $\mathcal{T}_k$ .
  - 8:     **end while**
  - 9:     Calculate optimal  $\alpha_k$  and build TPSFEM  $s_k$  on mesh  $\mathcal{T}_k$ ; calculate RMSE of  $s_k$ .
  - 10:     $k \leftarrow k + 1$ .
  - 11: **end while**
- 

#### 4. Approximations of the Dirichlet boundary conditions

Both Neumann and Dirichlet boundaries were investigated in [25, 28] to analyze the TPSFEM. Dirichlet boundaries are chosen in this study since they are efficient and more stable for oscillatory surfaces near boundaries [14]. The Dirichlet boundary conditions of TPSFEM smoother  $s$  are defined using  $\mathbf{c}$ ,  $\mathbf{g}_1$ ,  $\mathbf{g}_2$ , and  $\mathbf{w}$  values evaluated at boundary nodes. They are used to build  $s$  and initialize values of new nodes on domain boundaries. Values of  $\mathbf{c}$ ,  $\mathbf{g}_1$ , and  $\mathbf{g}_2$  are calculated using function  $f$  and its first-order derivatives directly. Stals and Roberts [28] showed that Lagrange multiplier  $\mathbf{w}$  can be approximated using the second-order derivatives of  $f$  if  $\Delta f = 0$  near the boundaries.

Real-world data sets often do not contain boundary information and these values cannot be computed directly. Fang and Stals [14] replaced  $\mathbf{c}$  values as the average of the predictor values  $\mathbf{y}$  near

the boundaries and set  $\mathbf{g}_1 = 0$ ,  $\mathbf{g}_2 = 0$ , and  $\mathbf{w} = 0$  for the Dirichlet boundaries. While it works for data sets with smooth surfaces and similar  $\mathbf{y}$  values near the boundaries, it is not applicable in general situations. If boundary conditions are not sufficiently accurate, they may affect the accuracy of  $s$  and mislead the error indicators during adaptive refinement, especially when data points are close to the boundary.

#### 4.1. Approximations using TPS

Many RBFs like Gaussian, multiquadrics, and TPS are continuously differentiable and have been used to approximate the surface of scattered data and its derivatives [17]. The TPS has smoothing properties similar to the TPSFEM and flexibility for small data sets (e.g., hundreds of data points). Thus, it is chosen to approximate Dirichlet boundary conditions for the TPSFEM. Despite that the TPS may become unstable near boundaries as argued by [15], it can greatly improve the accuracy of the TPSFEM  $s$  near boundaries compared to [14]. The values of the TPSFEM interpolant are listed in Table 1 with corresponding functions and kernels of the TPS. Given two predictor values  $\mathbf{x}_i$  and  $\mathbf{x}_j$ ,  $\mathbf{x}_{i1}$  represents the first entry of  $i$ -th predictor value.

**Table 1.** TPSFEM values, model problem functions, and TPS kernels.

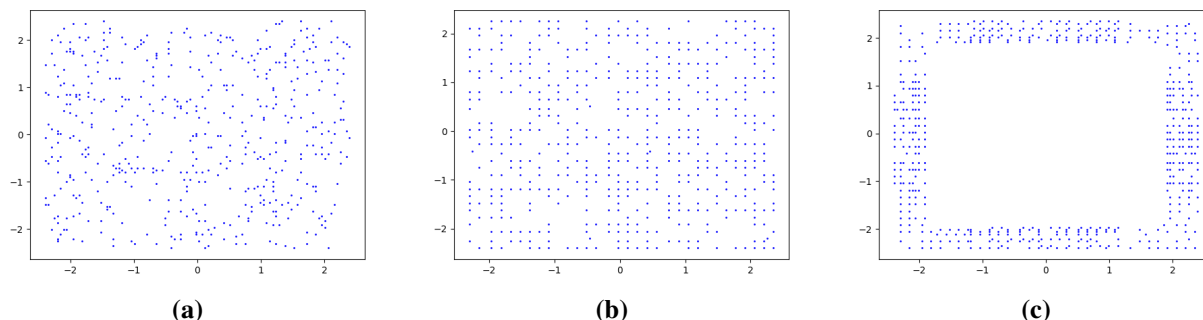
| TPSFEM                       | Function  | TPS Kernel  |
|------------------------------|---|---|
| $\mathbf{c}$                 | $f$   | $r^2 \log(r)$                                       |
| $\mathbf{g}_1$               | $\frac{\partial f}{\partial x_1}$   | $(2 \log r + 1)(\mathbf{x}_{i1} - \mathbf{x}_{j1})$ |
| $\mathbf{g}_2$               | $\frac{\partial f}{\partial x_2}$   | $(2 \log r + 1)(\mathbf{x}_{i2} - \mathbf{x}_{j2})$ |
| $-\frac{\mathbf{w}}{\alpha}$ | $\frac{\partial^2 f}{\partial x_1 x_1} + \frac{\partial^2 f}{\partial x_2 x_2}$ | $-\log r - 4$                                       |

The accuracy of the TPS  $t$  depends on a set of factors, including the choice of kernel, number  $\hat{n}$ , and distribution of the sampled data. The number of sampled points  $\hat{n}$  should be chosen as a trade-off between accuracy and cost and the sampling strategy may have large impacts especially when the sample size is small. A numerical experiment with noisy data is conducted to determine the optimal number and sampling strategy for  $t$ . Sampling data points randomly across the domain may lead to a large distance between neighboring points as shown in Figure 5(a) and is not considered for the experiment. Alternatively, one may use quadtrees, which place rectangular meshes over the domain and randomly select data points from small rectangles. This helps to bound the maximum distance between neighboring sampled points as illustrated in Figure 5(b). Regions of sampling may also be restricted to areas near boundaries like in Figure 5(c) to improve its accuracy near boundaries. These two approaches are evaluated in the experiments described in Supplementary C.

The RMSE of approximations for  $f$  and its first-order derivatives is initially reduced by increasing the number of sampled points. After the number reaches about 350, the RMSE for  $f$ ,  $\frac{\partial f}{\partial x_1}$ , and  $\frac{\partial f}{\partial x_2}$  using the boundary sampling cannot be reduced with more sampled points and the RMSE for those using uniform sampling is reduced by less than 6%, as discussed in Supplementary C. In addition, the second-order derivatives cannot be approximated accurately. While the boundary sampling performs relatively better than quadtree sampling in the experiment, the difference in RMSE gradually diminishes as  $\hat{n}$  increases. Therefore, 300 data points are sampled to build the TPS using quadtrees for the numerical



experiments in Section 5.

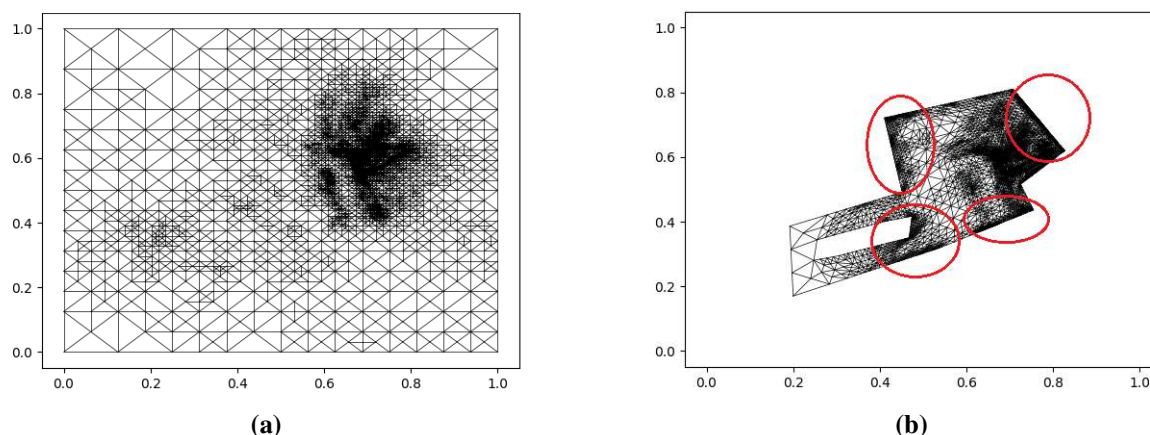


**Figure 5.** Distribution of 500 data points within the  $[-2.4, 2.4]^2$  region sampled (a) randomly; (b) using quadtrees; and (c) using quadtrees outside the  $[-1.9, 1.9]^2$  region.

#### 4.2. Over-refinement near boundaries

Over-refinement is an undesired behavior of adaptive refinement, where elements in certain regions are refined excessively and contribute little to improving the accuracy of the solution. This behavior often occurs near abrupt changes like discontinuities, where finer elements may not increase the accuracy but still indicate high errors. Fang and Stals [14] showed that over-refinement also appears in oscillatory regions near boundaries and the resulting adaptive meshes underperform compared to uniform meshes. This weakens the stability of the TPSFEM and adaptive refinement.

While the TPS approximations provide more accurate boundary conditions to build the TPSFEM smoother  $s$  like step 1 of Algorithm 1, it may be insufficient for adaptive refinement. An example is shown in Figure 6, where two adaptive meshes are generated for the same data set in different domains and Dirichlet boundary conditions are approximated using the TPS. The finer elements of a square domain in Figure 6(a) concentrate on oscillatory regions at the center of the domain. In contrast, the FE mesh in Figure 6(b) is constructed in an irregular domain and excessive refinement occurs near the boundaries. They contribute little to improving the accuracy of the TPSFEM, which leads to lower efficiency.



**Figure 6.** FE meshes generated for river data using a recovery-based error indicator in (a) a square domain; and (b) an irregular domain. Four regions with over-refinement of FEs near boundaries are marked by red circles.

Further investigations showed that most over-refinement occurs at step 6 of Algorithm 1, where new nodes are initialized during adaptive refinement. For adaptive refinement of FE approximations for PDEs, if a new node is added to the Dirichlet boundaries of FE meshes, its values are defined using known boundary functions like  $f$ . If a new interior node is added, its values are set as the average of neighboring nodes. Since boundary conditions may be unknown for the TPSFEM, they must be approximated. Supplementary C shows that the RMSE stops decreasing as the number of basis functions reaches a certain number due to noise in the data. The error indicators of the TPSFEM often rely on local information like a few data points or FEs and are sensitive to small changes. While the approximated boundary conditions are greatly improved compared to the constant values used in [14], they may either be too expensive or insufficiently accurate for error indicators. This leads to oscillatory surfaces near boundaries, which may produce high error indicator values and excessive refinement of FEs near boundaries.

An alternative for initializing new boundary nodes is proposed to avoid over-refinement during adaptive refinement. Instead of approximating new boundary node values during adaptive refinement, they are calculated as averaged values of their neighboring nodes. Thus, surfaces near boundaries will not be altered and remain smooth, which does not indicate high errors. Note that the new boundary conditions are not more accurate and the accuracy of  $s$  will not be improved in this process. Meanwhile, the TPS is still used to approximate Dirichlet boundary conditions to build  $s$  for irregular domains.

These two approaches are tested and compared in the numerical experiments to evaluate their effectiveness. The original approach that uses TPS approximations to provide boundary conditions for  $s$  and initialize new boundary node values during adaptive refinement is referred to as *TPS approximation*. The alternative that uses TPS approximations to build  $s$  and initialize new boundary node values using averaging is called *nodal average*.

## 5. Numerical experiments

The numerical experiments are conducted to evaluate the performance of the TPSFEM and adaptive refinement in both square and irregular domains. The three geological surveys for the experiments are from the United States Geological Survey\* and more details are given in Section 5.1. The settings of the experiments are provided in Section 5.2.

### 5.1. Real-world data

The first geological survey is an airborne magnetic survey collected from the Iron Mountain Menominee region of Michigan and Wisconsin [10]. Precambrian rocks in the region are poorly mapped and the detailed high-resolution airborne magnetic survey is used to better understand its lithology and structure. It comprises 1,011,468 data points and variables including latitude, longitude, and total magnetic field are selected for this study. The variations of magnetization are related to differences in rock types.

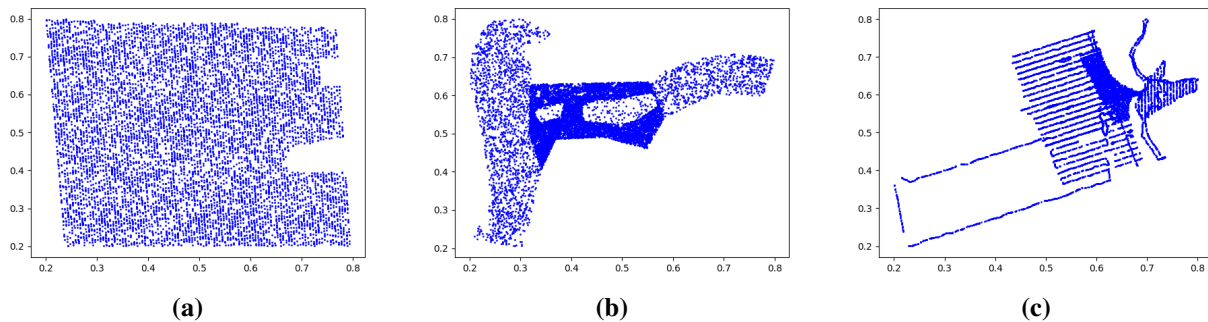
The second survey contains ground observations collected by terrestrial laser scanners in Grapevine Canyon near Scotty's Castle in Death Valley National Park [22]. It updated flood-inundation maps that were affected by a large flood in 2015 to current conditions. The observed data has been filtered of extraneous data such as vegetation, fences, power lines, and atmospheric interference for a continuous

---

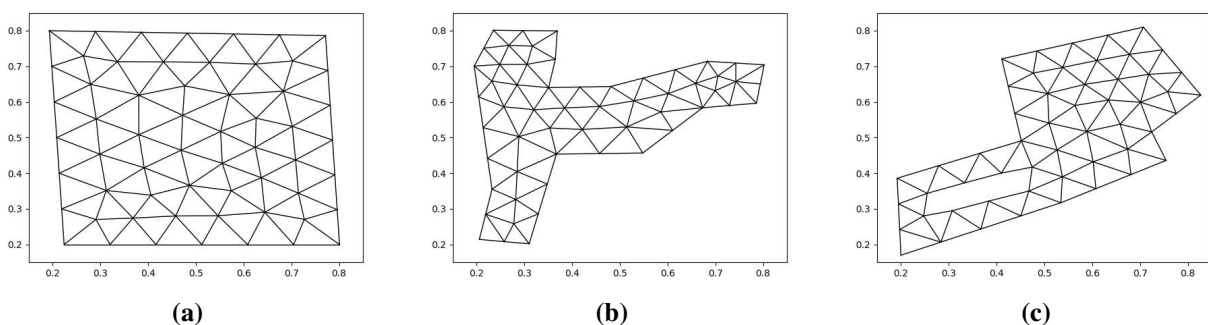
\*U.S. Geological Survey, <https://www.usgs.gov/>

model. The resulting 122,104 data points, including latitude, longitude, and elevation, are used to produce the DTM of the area.

The third survey is a nearshore bathymetric survey sampled at the mouth of the Unalakleet River in Alaska [26]. The survey comprises 895,748 data points consisting of latitude, longitude, and height in meters regarding the reference ellipsoid. The data recorded depths from the seafloor to the echosounder and was collected using a small boat equipped with a single-beam sonar system.



**Figure 7.** Distributions of a subset of randomly selected  $x$  variables in (a) mountain; (b) canyon; and (c) river data sets.



**Figure 8.** Irregular FE meshes for (a) mountain; (b) canyon; and (c) river data sets.

The data from these three surveys will be referred to as the Mountain data, Canyon data, and River data in the rest of the article. They are chosen due to their distinct data distribution patterns as illustrated in Figure 7. Corresponding initial irregular FE meshes are shown in Figure 8. The Mountain data was collected using an unmanned aerial vehicle in transects, which are relatively uniform across the domain as shown in Figure 7(a). In contrast, the Canyon data in Figure 7(b) are limited by terrains and equipment. Sampled points of the ground observations concentrate on the bottom and steep sides of the deep canyon that may be altered by the flood. More observations are located at the center, which leads to uneven data densities. The River data in Figure 7(c) reflects routes of a boat that form frames with various sparseness. It also contains a region without any data points corresponding to the interior hole in the bottom left of the FE mesh in Figure 8(c). More data points were sampled around the river mouth around point  $[0.65, 0.6]$  to model complex river beds.

## 5.2. Experimental design

Recall that we consider both square and irregular domains. All initial FE meshes in square domains contain 25 nodes and are refined using uniform or adaptive refinement for at most 10 and 8 iterations, respectively. The initial irregular FE meshes for the three data sets are generated using alphashape and pymesh as shown in Figure 8 and contain 59, 56, and 53 nodes, respectively. They are refined using uniform or adaptive refinement for 8 or 7 iterations, respectively. The final square and irregular domains have similar numbers of nodes, which allows us to compare their performance. The predictor values  $\mathbf{x}$  of the three data sets are scaled and fitted inside the  $[0.2, 0.8]^2$  region of a square  $[0, 1]^2$  domain. The response values  $\mathbf{y}$  are sampled on different scales and normalized in the range of  $[0, 1]$  for comparison.

The efficiency of  $s$  is measured using the RMSE versus the number of nodes in the meshes, where  $\text{RMSE} = \left(\frac{1}{n} \sum_{i=1}^n (s(\mathbf{x}_i) - y_i)^2\right)^{\frac{1}{2}}$ . Additionally, the ratio of interior nodes within the radius of 0.005 of the boundary nodes indicates the degree of over-refinement near the boundaries. This radius is chosen such that the ratio only includes nodes generated during over-refinement near the boundaries. Note that the lengths of more than 50% of edges in the final adaptive FE meshes are shorter than 0.005.

Their performance is compared through convergence plots and tables containing statistics of final meshes generated using different approaches. The plots display the convergence of RMSE in logarithmic scales for the three data sets using both uniform or adaptive refinement with the auxiliary problem and recovery-based error indicators described in Supplementary B. Their performance is evaluated in square and irregular domains with the TPS approximation or nodal average. Additional metrics of the final meshes are provided in tables, including the number of nodes, time for solving Eq (3.2) of the TPSFEM, RMSE, and maximum absolute residual errors (MAX). The computational costs, including time for building and solving systems, error indicators, and GCV, of the TPSFEM are affected by factors like the number of nodes in FE meshes, and the number and distribution patterns of data points. A detailed analysis is too complicated as discussed in Fang and Stals [14] and not the focus of this article. Hence, we only include system-solving time, which reflects the difference in their efficiency.

The experiments are conducted using a TPSFEM program<sup>†</sup> implemented in Python 3.8. The systems are solved by a sparse solver (PyPardiso<sup>‡</sup>) and the time is measured in seconds. All experiments are conducted on a personal computer with a Microsoft Windows 10 operating system and an Intel<sup>®</sup> Core<sup>™</sup> i9-10900 CPU @ 2.80 GHz configuration with 64.0 GB of RAM.

## 6. Results and discussions

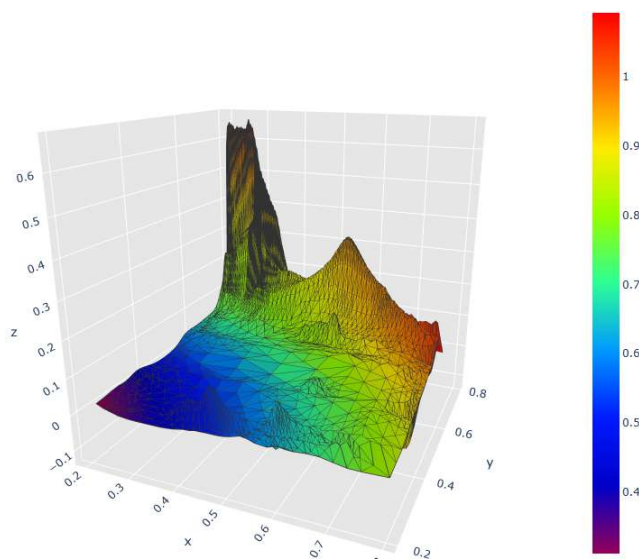
The numerical results and discussions for the Mountain, Canyon, and River data are provided in Sections 6.1, 6.2, and 6.3, respectively. A comparison of the TPSFEM to the TPS and two compactly supported RBFs (CSRBFs) is given in Section 6.4.

<sup>†</sup>TPSFEM program, <https://bitbucket.org/fanglishan/tpsfem-program/>

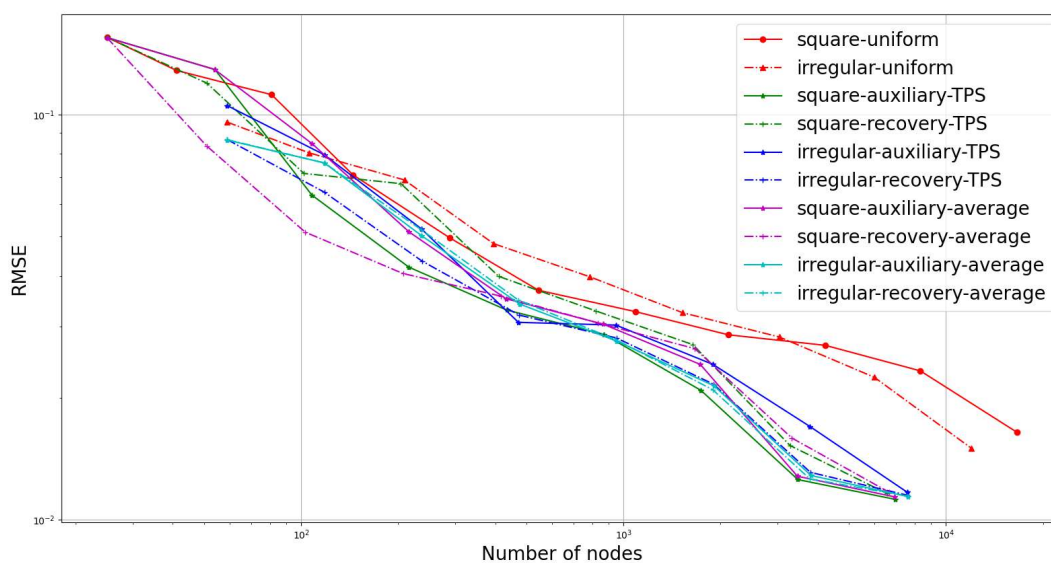
<sup>‡</sup>PyPardiso, <https://pypi.org/project/pypardiso/>

### 6.1. Mountain data

A triangulated surface plot for the Mountain data generated using the recovery-based error indicator in an irregular domain is shown in Figure 9. The Mountain data consists of several oscillatory surfaces at the upper left corner and relatively smooth surfaces in the rest of the domain. High peaks near  $[0.25, 0.8]$  show high magnetic field readings and suggest potential mineral resources. Adaptive refinement accurately detects oscillatory regions and iteratively refines them to achieve the required accuracy. The resulting mesh contains markedly finer elements over the peaks and coarser elements elsewhere, which effectively reduces computational costs and memory requirements of the TPSFEM as discussed below.



**Figure 9.** Triangulated surface of TPSFEM for Mountain data in an irregular domain.



**Figure 10.** RMSE of TPSFEM for Mountain data.

The convergence of the RMSE of the TPSFEM for the Mountain data is shown in Figure 10 and metrics of the final meshes are listed in Table 2. The convergence curves correspond to TPSFEM in square or irregular domains with meshes refined uniformly or adaptively using the auxiliary problem or recovery-based error indicators. If the values of the new boundary nodes are approximated using the TPS during adaptive refinement, the curves are marked as TPS. Otherwise, if they are set as the average of their neighboring nodes, the curves are marked as average.

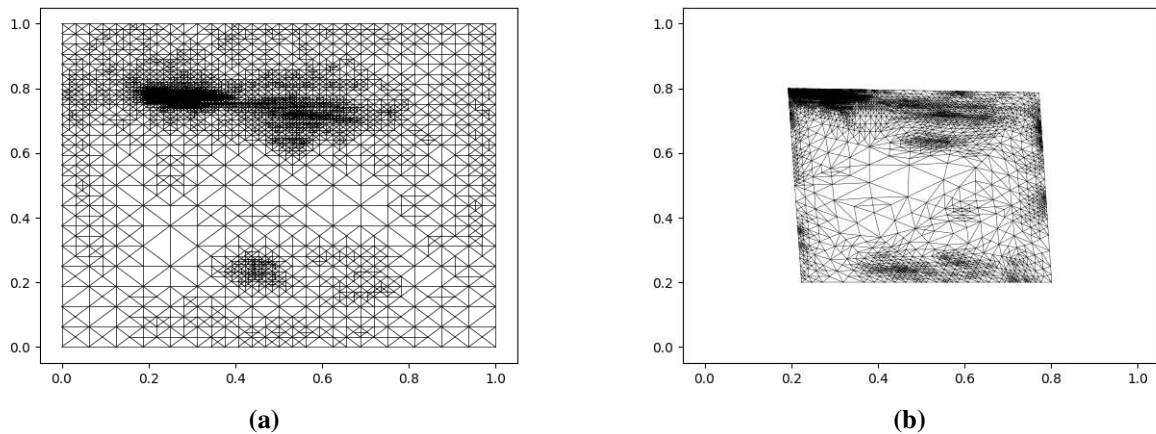
**Table 2.** Metrics of TPSFEM for Mountain data.

| Domain    | Metric  | TPS approximation |           |          | Average   |          |
|-----------|---------|-------------------|-----------|----------|-----------|----------|
|           |         | Uniform           | Auxiliary | Recovery | Auxiliary | Recovery |
| Square    | # nodes | 16641             | 6948      | 6564     | 6923      | 6656     |
|           | Time    | 0.504             | 0.225     | 0.226    | 0.299     | 0.216    |
|           | RMSE    | 0.0164            | 0.0112    | 0.0116   | 0.0114    | 0.0117   |
|           | MAX     | 0.45              | 0.353     | 0.356    | 0.356     | 0.355    |
| Irregular | # nodes | 11969             | 7576      | 7600     | 7634      | 7560     |
|           | Time    | 0.353             | 0.247     | 0.245    | 0.268     | 0.232    |
|           | RMSE    | 0.0150            | 0.0117    | 0.0115   | 0.0114    | 0.0114   |
|           | MAX     | 0.441             | 0.356     | 0.352    | 0.355     | 0.354    |

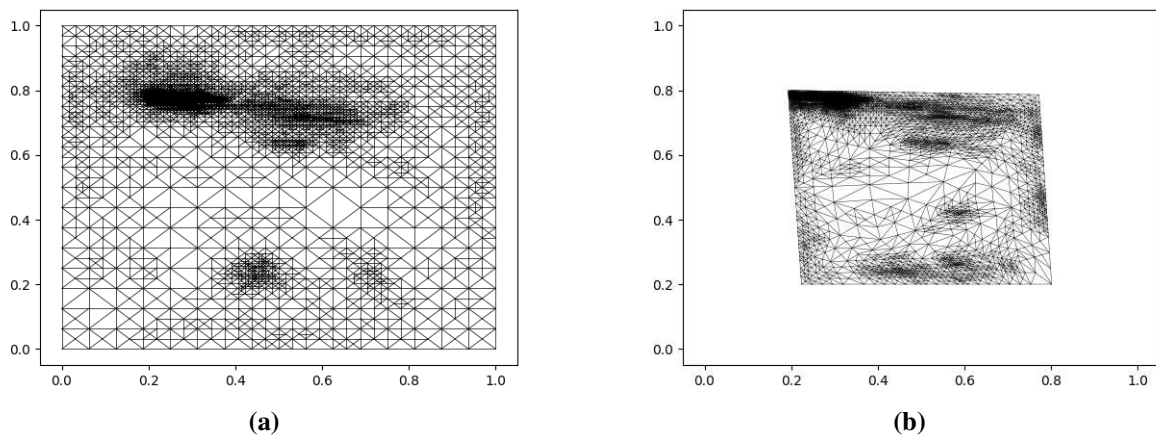
Adaptive refinement outperforms uniform refinement for the Mountain data in both square and irregular domains. Since most interpolated surfaces are smooth, the TPSFEM can achieve high accuracy with relatively coarse elements. The two error indicators have similar performance. Without adaptive refinement, the TPSFEM in irregular domains achieves lower RMSE with fewer nodes than square domains. The shorter distance between data and boundaries reduces the number of elements that do not cover any data points, which leads to higher efficiency.

When values of boundary nodes are approximated using the TPS, adaptive refinement achieves lower efficiency in irregular domains than in square domains. Adaptive meshes in irregular domains using the recovery-based error indicator achieve similar RMSE with 15.78% more nodes than in square domains. The mesh refined using the auxiliary problem has higher RMSE and more nodes, which is markedly more inefficient. Two FE meshes generated using the recovery-based error indicator are shown in Figure 11. The FE mesh in a square domain has 0% of the interior nodes near the boundaries. In contrast, the adaptive mesh in the irregular domain in Figure 11(b) has fine elements near boundaries and about 8.32% of interior nodes close to boundaries.

When new boundary node values are set using nodal average during adaptive refinement, over-refinement near the boundary is reduced and fewer FEs near boundaries are found in Figure 12(b) compared to Figure 11(b). The ratio of interior nodes near boundaries is reduced from 8.32% to about 2.22% in Figure 12(b). However, no significant difference is observed between FE meshes in square domains as shown in Figures 11(a) and 12(a). Since most of the Mountain data's oscillatory regions are near the boundary, the reduction of over-refinement in Figure 12(b) does not have marked impacts on the RMSE. Another interesting finding is that the auxiliary problem error indicator achieves the lowest RMSE in square domains with new boundary node values approximated using the TPS. Adaptive refinement identifies regions that do not require fine FEs and obtains similar efficiency.



**Figure 11.** FE meshes in (a) a square domain; and (b) an irregular domain. New boundary node values are approximated using TPS.



**Figure 12.** FE meshes in (a) a square domain; and (b) an irregular domain. New boundary node values are set as the average of neighboring nodes.

## 6.2. Canyon data

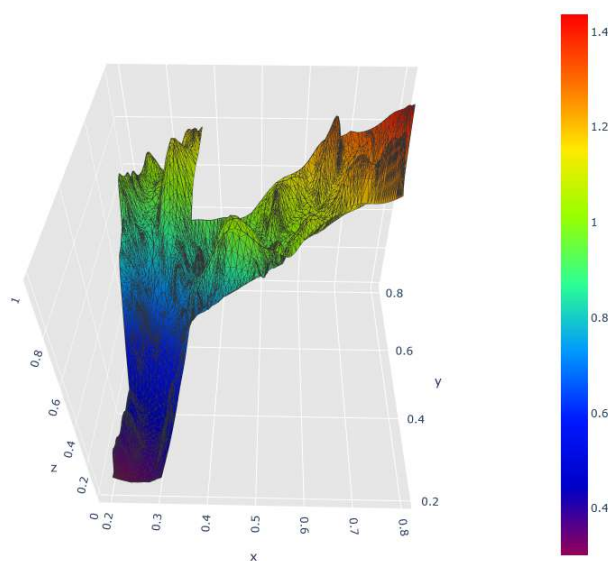
A triangulated surface plot for the Canyon data generated using the auxiliary problem error indicator in an irregular domain is shown in Figure 13. The Canyon data consists of steep sides and a gradual bottom of the observed canyon. The steep sides are near the exterior of the Canyon data and these regions require finer elements to model. In contrast, the bottom of the canyon and regions without data points only need coarse elements.

The convergence of the RMSE of the TPSFEM for the Canyon data is shown in Figure 14 and the metrics of the final meshes are shown in Table 3. It is shown that adaptive meshes significantly improve the efficiency of the Canyon data in square domains with new boundary node values calculated by the TPS. For example, the adaptive mesh of the auxiliary problem error indicator achieves about 15.74% of RMSE using 39.74% of nodes compared to the uniform mesh. Adaptive refinement produces finer elements on steep sides and coarser elements on regions without data. This allows the TPSFEM to achieve high RMSE while retaining efficiency. The auxiliary problem error indicator outperforms the recovery-based error indicator in all cases of Table 3 as it better detects oscillatory behavior using local

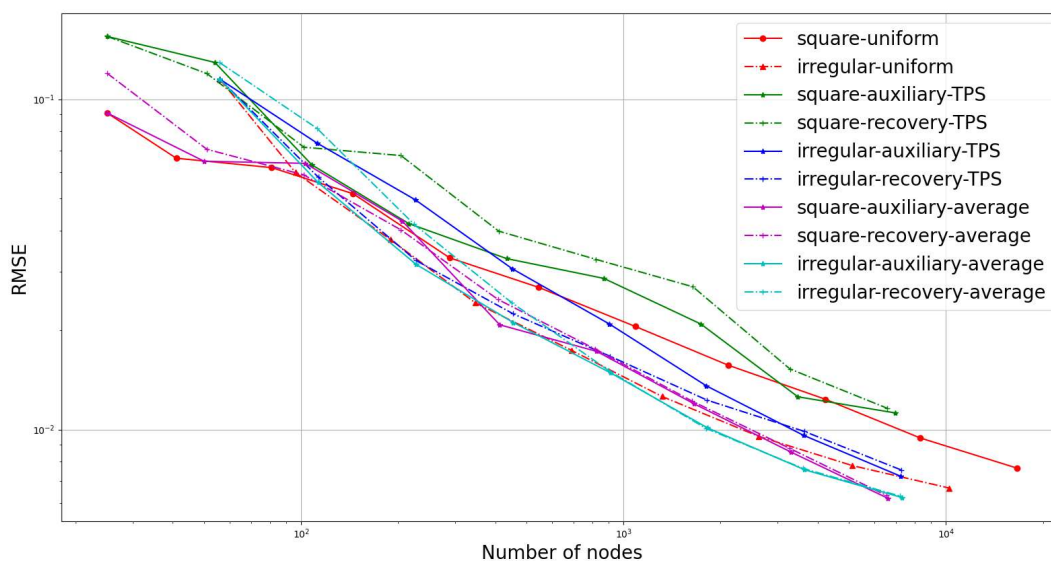


data points.

Since the Canyon data only covers a small portion of the domain as shown in Figure 7, irregular domains significantly improve the TPSFEM's efficiency. However, adaptive refinement fails to improve the efficiency in irregular domains when new boundary node values are initialized using the TPS approximations. An adaptive mesh generated using the auxiliary problem error indicator is shown in Figure 15(a), where severe over-refinement is observed near boundaries and about 33.43% of interior nodes are close to boundaries. Consequently, the RMSE for both of the error indicators in irregular domains is significantly higher, as well as the numbers of nodes, than those in square domains. Over-refinement near boundaries has a major impact on the Canyon data compared to the Mountain data.



**Figure 13.** Triangulated surface of TPSFEM for Canyon data in an irregular domain.

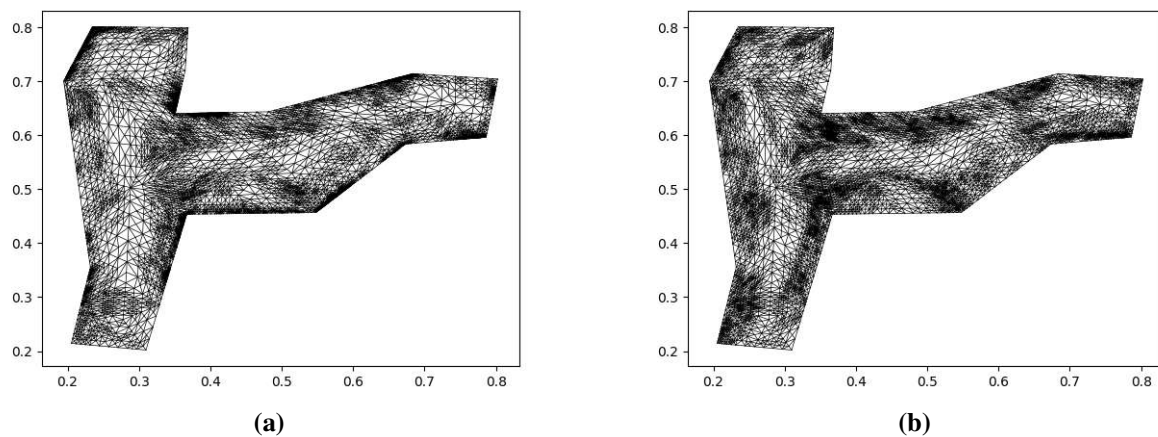


**Figure 14.** RMSE of TPSFEM for Canyon data.



**Table 3.** Metrics of TPSFEM for Canyon data.

| Domain    | Metric  | TPS approximation |           |          | Average   |          |
|-----------|---------|-------------------|-----------|----------|-----------|----------|
|           |         | Uniform           | Auxiliary | Recovery | Auxiliary | Recovery |
| Square    | # nodes | 16641             | 6612      | 6528     | 6608      | 6534     |
|           | Time    | 0.448             | 0.231     | 0.223    | 0.228     | 0.235    |
|           | RMSE    | 0.0394            | 0.00620   | 0.00665  | 0.00619   | 0.00632  |
|           | MAX     | 0.0524            | 0.0623    | 0.0650   | 0.0656    | 0.0605   |
| Irregular | # nodes | 10241             | 7238      | 7265     | 7294      | 7204     |
|           | Time    | 0.299             | 0.236     | 0.242    | 0.232     | 0.229    |
|           | RMSE    | 0.00665           | 0.00722   | 0.00754  | 0.00623   | 0.00629  |
|           | MAX     | 0.118             | 0.117     | 0.116    | 0.116     | 0.117    |

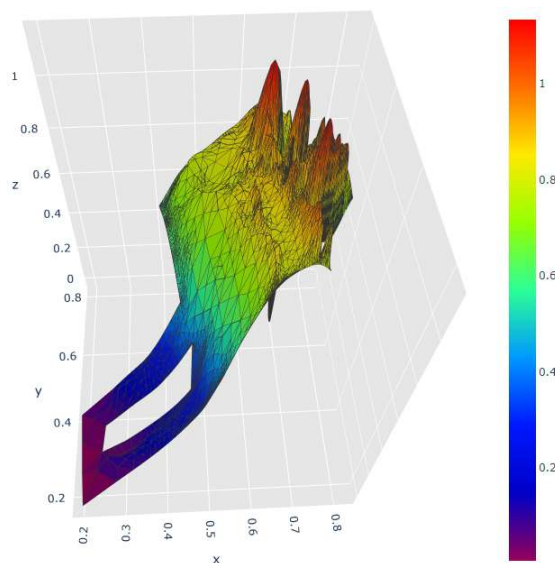
**Figure 15.** FE meshes in an irregular domain generated with new boundary node values (a) approximated using TPS; and (b) set as the average of neighboring nodes.

After adopting the nodal-average approach, over-refinement near boundaries is markedly reduced as shown in Figure 15(b) and the ratio of interior nodes near boundaries is reduced from 33.43% to about 5.61%. Consequently, the RMSE of final adaptive meshes in both square and irregular domains achieves similar RMSE as those in square domains. In addition, the TPSFEM has much higher MAX in irregular domains than in square domains. Since the Canyon data has more oscillatory surfaces near boundaries than the other two data sets, the approximated boundary conditions have larger impacts on  $s$ , especially near boundaries.

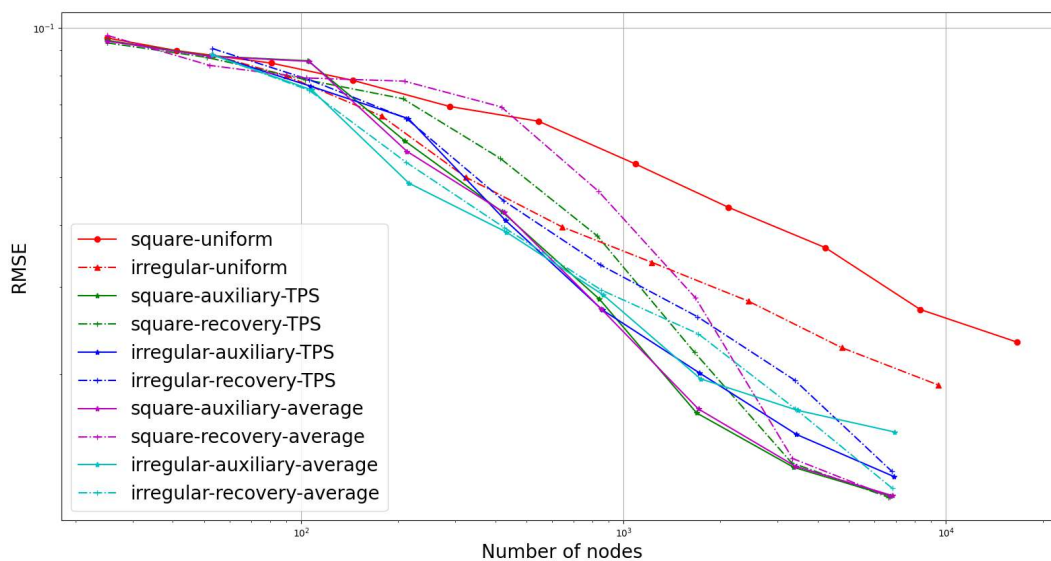
### 6.3. River data

A triangulated surface plot for the River data generated using the recovery-based error indicator in its irregular domain is shown in Figure 16. The River data is the most complicated data set with various routes and data densities as described in Section 5.1. The sampled data concentrates at the river mouth, where currents change and river beds are eroded by the flow. This leads to oscillatory surfaces and deep holes under the river mouth near point  $[0.8, 0.6]$ . The other regions are relatively smooth and only require coarse elements. Only elements near the river mouth are refined with finer elements.

The convergence of the RMSE of the TPSFEM for the River data is shown in Figure 17 and metrics of the final meshes are shown in Table 4. Adaptive meshes in square domains achieve much higher convergence rates compared to uniform meshes for the River data. Since the river mouth region is significantly more oscillatory than other regions, adaptive meshes are more efficient than uniform meshes similar to the Canyon data. Both of the error indicators have similar performance in this data set.



**Figure 16.** Triangulated surface of TPSFEM for River data in an irregular domain.

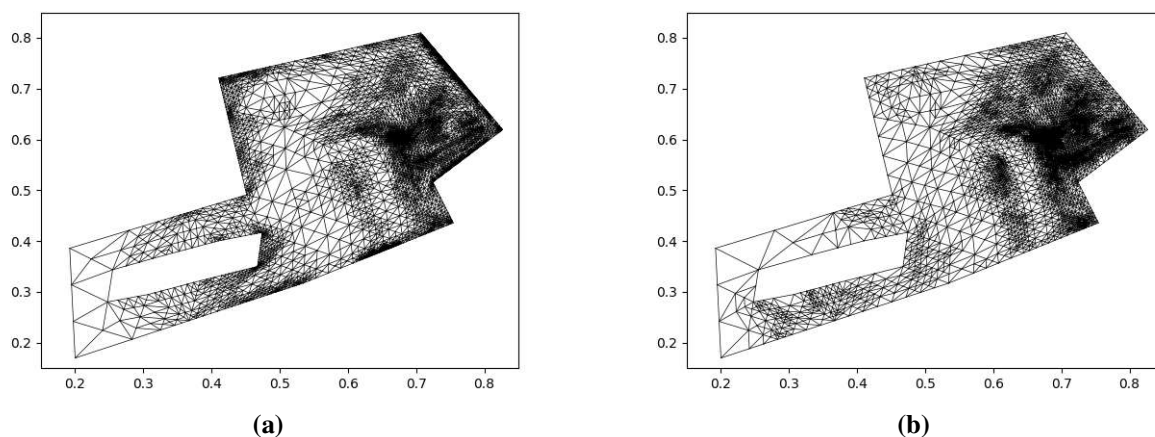


**Figure 17.** RMSE of TPSFEM for River data.

**Table 4.** Metrics of TPSFEM for River data.

| Domain    | Metric  | TPS approximation |           |          | Average   |          |
|-----------|---------|-------------------|-----------|----------|-----------|----------|
|           |         | Uniform           | Auxiliary | Recovery | Auxiliary | Recovery |
| Square    | # nodes | 16641             | 6740      | 6642     | 6828      | 6688     |
|           | Time    | 0.465             | 0.234     | 0.228    | 0.227     | 0.248    |
|           | RMSE    | 0.0232            | 0.0113    | 0.0113   | 0.0114    | 0.0113   |
|           | MAX     | 0.379             | 0.194     | 0.193    | 0.199     | 0.193    |
| Irregular | # nodes | 9488              | 6928      | 6792     | 6880      | 6816     |
|           | Time    | 0.304             | 0.238     | 0.243    | 0.230     | 0.240    |
|           | RMSE    | 0.0190            | 0.0152    | 0.0127   | 0.0124    | 0.0118   |
|           | MAX     | 0.283             | 0.204     | 0.174    | 0.172     | 0.172    |

The TPSFEM in irregular domains with uniform refinement obtains 81.90% of RMSE using 57.02% of the number of nodes compared to that in square domains. The improvement is not as significant as adaptive refinement, which achieves 48.88% of RMSE using 40.50% of the number of nodes in square domains. Adaptive refinement also improves the MAX in all test cases. In addition, adaptive refinement using the TPS approximations leads to higher RMSE and number of nodes in irregular domains compared to square domains. This is caused by small oscillatory regions surrounding the river mouth. Since the area is close to the boundaries of the irregular domain, the error indicators are misled and over-refined in multiple regions near the boundaries of the FE mesh as illustrated in Figure 18(a). Over-refinement is significantly reduced using the nodal-average approach as shown in Figure 18(b), where only a small amount of over-refinement is observed. The ratio of interior nodes near boundaries has also been drastically reduced from about 13.18% to 1.19%.



**Figure 18.** FE meshes in an irregular domain generated with values of the new boundary nodes (a) approximated using TPS; and (b) set as the average of neighboring nodes.

#### 6.4. Comparison to RBFs

The performance of the TPSFEM is compared to the TPS and two CSRBFs from [29] and [5] to illustrate its effectiveness. Let  $\bar{x}_i$  be a control point, where  $1 \leq i \leq \bar{n}$ . Its kernel is defined as  $\phi_i(r)$  in

Table 5, where  $r = \|\bar{\mathbf{x}}_i - \mathbf{x}\|_2$  is the Euclidean distance between  $\bar{\mathbf{x}}_i$  and  $\mathbf{x}$ . The TPS has global support and the two CSRBFs have compact support as their kernels  $\phi(r)$  are defined as  $\phi(r) = 0$  if  $r > 1$ . The kernels of the two CSRBFs are scaled using a radius of support  $\rho$  and their values are computed as  $\phi(r/\rho)$ , which markedly reduces the computational costs. In the experiments, two values of  $\rho$  are chosen for each CSRBF such that the compact support of each  $\bar{\mathbf{x}}_i$  covers a fixed number of data points [9]. For example, the CSRBFs for the Mountain data with a radius of support 0.0691 and 0.120 cover approximately 100 or 300 sampled points, respectively.

**Table 5.** Kernels of TPS and CSRBFs.

| Technique | Kernel $\phi$  |
|-----------|--|
| TPS       | $r^2 \log(r)$  |
| Buhmann   | $1/15 + 19/6r^2 - 16/3r^3 + 3r^4 - 16/15r^5 + 1/6r^6 + 2r^2 \log(r)$ |
| Wendland  | $(1 - r)^4(4r + 1)$  |

The control points of the RBFs are sampled by finding data points close to nodes of a uniform rectangular mesh over the domain. Given a mesh with mesh size  $\bar{h}$ , regions where the distance between the nearest nodes and data points is greater than  $\bar{h}/3$  are ignored. Thus, the maximum and minimum distances between neighboring control points are bounded. The smoothing parameters of the RBFs are calculated using the GCV.

The efficiency of the RBFs and TPSFEM are compared using statistics listed in Table 6. The RBFs have  $\bar{n}$  basis functions and the TPSFEM has  $m$  nodes and one basis function for each node. The computational costs are evaluated using the ratio and numbers of nonzero entries in the system of equations and the time to solve the system, which is measured in seconds. The systems of the TPS are solved using a direct solver and the systems of the CSRBFs and TPSFEM are solved using the sparse solver (PyPardiso). The TPSFEM generated in irregular domains using the recovery-based error indicator and nodal average for the three data sets have also been listed in Table 6 for comparison.

As shown in Table 6, the CSRBFs from Buhmann and Wendland have similar RMSE and MAX to the TPS for the three data sets. Since the systems of the CSRBFs are much sparser than the dense systems of the TPS as shown by the nonzero ratios, the CSRBFs require significantly less time to solve the system. The CSRBF of Buhmann achieves higher efficiency for the Mountain and Canyon data, but it underperforms for the River data compared to the CSRBF of Wendland. The TPSFEM with uniform meshes in square domains are markedly less efficient compared to the CSRBFs as shown in Tables 2, 3, and 4. However, the TPSFEM with adaptive refinement achieves much lower RMSE and MAX for all three data sets. Additionally, their systems are much sparser and take less time to solve.

The performance of the CSRBFs is affected by factors like the radius of support  $\rho$ . Table 6 contains attributes of the CSRBFs with  $\rho$  that cover about 100 and 300 sample points corresponding to columns 4 and 6 and rows 5 and 7, respectively. While the increase in  $\rho$  may reduce the RMSE and MAX of the CSRBFs in other studies, it showed little improvement for the three data sets. Note that the performance of the CSRBFs can be improved using a localized radius of support, preconditioners, or parallel solvers [9]. Nevertheless, this experiment demonstrates the TPSFEM's effectiveness.

**Table 6.** Metrics of CSRBF for Mountain, Canyon, and River data.

| Data     | Metric    | TPS        | Buhmann   |            | Wendland  |            | TPSFEM  |
|----------|-----------|------------|-----------|------------|-----------|------------|---------|
| Mountain | Radius    |            | 0.0691    | 0.120      | 0.0691    | 0.120      |         |
|          | # basis   | 6673       | 6673      | 6673       | 6673      | 6673       | 7560    |
|          | # nonzero | 44,528,929 | 1,809,811 | 4,959,647  | 1,809,811 | 4,959,64   | 426,448 |
|          | Ratio     | 100%       | 4.06%     | 11.1%      | 4.06%     | 11.1%      | 0.044%  |
|          | Time      | 2.56       | 0.777     | 1.38       | 0.779     | 1.45       | 0.232   |
|          | RMSE      | 0.0166     | 0.0165    | 0.0165     | 0.0171    | 0.0172     | 0.0114  |
|          | MAX       | 0.575      | 0.573     | 0.573      | 0.591     | 0.591      | 0.354   |
| Canyon   | Radius    |            | 0.0684    | 0.118      | 0.0684    | 0.118      |         |
|          | # basis   | 6807       | 6807      | 6807       | 6807      | 6807       | 7204    |
|          | # nonzero | 46,335,249 | 5,984,357 | 10,450,523 | 5,984,357 | 10,450,523 | 409,826 |
|          | Ratio     | 100%       | 12.9%     | 22.6%      | 12.9%     | 22.6%      | 0.046%  |
|          | Time      | 3.03       | 1.48      | 2.35       | 1.42      | 2.41       | 0.229   |
|          | RMSE      | 0.00658    | 0.00668   | 0.00658    | 0.00680   | 0.00679    | 0.00629 |
|          | MAX       | 0.0808     | 0.0795    | 0.0798     | 0.0914    | 0.0904     | 0.117   |
| River    | Radius    |            | 0.0699    | 0.121      | 0.0699    | 0.121      |         |
|          | # basis   | 6517       | 6517      | 6517       | 6517      | 6517       | 6816    |
|          | # nonzero | 42,471,289 | 4,240,983 | 9,815,249  | 4,240,983 | 9,815,249  | 399,667 |
|          | Ratio     | 100%       | 9.97%     | 23.2%      | 9.97%     | 23.2%      | 0.05%   |
|          | Time      | 2.07       | 1.16      | 2.36       | 1.15      | 2.35       | 0.240   |
|          | RMSE      | 0.0144     | 0.0146    | 0.0145     | 0.0139    | 0.0140     | 0.0118  |
|          | MAX       | 0.370      | 0.373     | 0.372      | 0.344     | 0.342      | 0.172   |

## 7. Conclusions

We study the TPSFEM and adaptive mesh refinement in irregular domains with complex geometric shapes, which better represent the geometry for the domain of interest. Unknown Dirichlet boundary conditions for real-world data sets are approximated using the TPS with a small subset of sampled data points. They improve the accuracy of the TPSFEM for irregular domains with data points close to boundaries. Over-refinement of FEs near boundaries weakens the performance of adaptive refinement for the TPSFEM. An alternative is proposed to initialize new boundary node values as the average of neighboring nodes. This improves the stability of adaptive refinement when exact boundary conditions are unavailable.

Numerical experiments are conducted to investigate the performance of the TPSFEM and adaptive refinement in square and irregular domains. The experiments use three geological surveys with varied distribution patterns and irregular domains. Two approaches to initializing new boundary node values during adaptive refinement are compared. The results show that adaptive refinement markedly improves the efficiency of the TPSFEM for all three surveys in both square and irregular domains. While over-refinement of FEs occurs in irregular domains, it is significantly reduced using the proposed averaging initialization for new boundary nodes. Irregular domains also increase the

efficiency of the TPSFEM with uniform refinement, though adaptive refinement achieves slightly higher convergence in square domains. The recovery-based error indicator is more stable and slightly overperforms the auxiliary problem error indicator. The comparison of the TPSFEM, TPS, and two chosen CSRBFs shows that the TPSFEM achieves lower RMSE with costs. Overall, concerning efficiency and flexibility for digital terrain modeling in irregular domains, our findings show the TPSFEM is worthy of recommendation. Our future work will focus on extending it to three-dimensional data sets and improving its accuracy for surfaces with abrupt changes.

## Acknowledgments

I would like to express my great appreciation to Dr. Linda Stals for her valuable and constructive suggestions during the planning and development of this research work. Her willingness to give her time so generously has been very much appreciated.

This work was supported by the Natural Science Foundation of Xiamen [Grant No. 3502Z202371014] and the Scientific Research Funds of Huaqiao University.

## Conflict of interest

The author declares no conflict of interest.

## References

1. M. Bartels, H. Wei, Towards DTM generation from LiDAR data in hilly terrain using wavelets, *4th international workshop on pattern recognition in remote sensing, in conjunction with international conference on pattern recognition*, **1** (2006), 33–36.
2. R. K. Beatson, J. B. Cherrie, C. T. Mouat, Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration, *Adv. Comput. Math.*, **11** (1999), 253–270. <https://doi.org/10.1023/A:1018932227617>
3. I. C. Briggs, Machine contouring using minimum curvature, *Geophysics*, **39** (1974), 39–48. <https://doi.org/10.1190/1.1440410>
4. P. Brufau, M. Vázquez-Cendón, P. García-Navarro, A numerical model for the flooding and drying of irregular domains, *Int. J. Numer. Methods Fluids*, **39** (2002), 247–275. <https://doi.org/10.1002/flid.285>
5. M. D. Buhmann, Radial functions on compact support, *Proc. Edinburgh Math. Soc.*, **41** (1998), 33–46. <https://doi.org/10.1017/S0013091500019416>
6. M. Căteanu, A. Ciubotaru, The effect of lidar sampling density on DTM accuracy for areas with heavy forest cover, *Forests*, **12** (2021), 265. <https://doi.org/10.3390/f12030265>
7. C. Chen, Y. Li, C. Yan, A random features-based method for interpolating digital terrain models with high efficiency, *Math. Geosci.*, **52** (2020), 191–212. <https://doi.org/10.1007/s11004-019-09801-z>

8. Z. M. Chen, R. Tuo, W. L. Zhang, Stochastic convergence of a nonconforming finite element method for the thin plate spline smoother for observational data, *SIAM J. Numer. Anal.*, **56** (2018), 635–659. <https://doi.org/10.1137/16M109630X>
9. S. Deparis, D. Forti, A. Quarteroni, A rescaled localized radial basis function interpolation on non-Cartesian and nonconforming grids, *SIAM J. Sci. Comput.*, **36** (2014), A2745–A2762. <https://doi.org/10.1137/130947179>
10. B. J. Drenth, Airborne magnetic total-field survey, Iron Mountain-Menominee region, Michigan-Wisconsin, USA, 2021. <https://doi.org/10.5066/F7W66J0N>
11. T. A. Driscoll, A. R. Heryudono, Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, *Comput. Math. Appl.*, **53** (2007), 927–939. <https://doi.org/10.1016/j.camwa.2006.06.005>
12. A. Falini, M. L. Sampoli, Adaptive refinement in advection–diffusion problems by anomaly detection: A numerical study, *Algorithms*, **14** (2021), 328. <https://doi.org/10.3390/a14110328>
13. L. Fang, L. Stals, Adaptive finite element thin-plate spline with different data distributions, In: *Domain Decomposition Methods in Science and Engineering XXVI*, Cham: Springer, 2023, 661–669. [https://doi.org/10.1007/978-3-030-95025-5\\_72](https://doi.org/10.1007/978-3-030-95025-5_72)
14. L. Fang, L. Stals, Data-based adaptive refinement of finite element thin plate spline, *J. Comput. Appl. Math.*, **451** (2024), 115975. <https://doi.org/10.1016/j.cam.2024.115975>
15. B. Fornberg, T. A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, *Comput. Math. Appl.*, **43** (2002), 473–490. [https://doi.org/10.1016/S0898-1221\(01\)00299-1](https://doi.org/10.1016/S0898-1221(01)00299-1)
16. E. Galin, E. Guérin, A. Peytavie, G. Cordonnier, M.-P. Cani, B. Benes, et al., A review of digital terrain modeling, *Comput. Graph. Forum*, **38** (2019), 553–577. <https://doi.org/10.1111/cgf.13657>
17. A. Karageorghis, C. S. Chen, Y. S. Smyrlis, A matrix decomposition RBF algorithm: Approximation of functions and their derivatives, *Appl. Numer. Math.*, **57** (2007), 304–319. <https://doi.org/10.1016/j.apnum.2006.03.028>
18. H. Laga, Y. Guo, H. Tabia, R. B. Fisher, M. Bennamoun, *3D Shape analysis: Fundamentals, theory, and applications*, John Wiley & Sons, 2018. <https://doi.org/10.1002/9781119405207>
19. R. A. McLaren, T. J. Kennie, Visualisation of digital terrain models: Techniques and applications, In: *Three Dimensional Applications in GIS*, London: CRC Press, 79–98. <https://doi.org/10.1201/9781003069454-6>
20. W. F. Mitchell, A comparison of adaptive refinement techniques for elliptic problems, *ACM TOMS*, **15** (1989), 326–347. <https://doi.org/10.1145/76909.76912>
21. A. L. Montealegre, M. T. Lamelas, J. De la Riva, Interpolation routines assessment in als-derived digital elevation models for forestry applications, *Remote Sens.*, **7** (2015), 8631–8654. <https://doi.org/10.3390/rs70708631>
22. C. M. Morris, T. L. Welborn, J. T. Minear, Geospatial data, tabular data, and surface-water model archive for delineation of flood-inundation areas in Grapevine Canyon near Scotty’s Castle, Death Valley National Park, California, 2020. <https://doi.org/10.5066/P9IPKW55>

23. A. Papastavrou, R. Verfürth, A posteriori error estimators for stationary convection–diffusion problems: A computational comparison, *Comput. Methods Appl. Mech. Eng.*, **189** (2000), 449–462. [https://doi.org/10.1016/S0045-7825\(99\)00301-1](https://doi.org/10.1016/S0045-7825(99)00301-1)
24. T. Ramsay, Spline smoothing over difficult regions, *J. R. Stat. Soc.: Ser. B*, **64** (2002), 307–319. <https://doi.org/10.1111/1467-9868.00339>
25. S. Roberts, M. Hegland, I. Altas, Approximation of a thin plate spline smoother using continuous piecewise polynomial functions, *SIAM J. Numer. Anal.*, **41** (2003), 208–234. <https://doi.org/10.1137/S0036142901383296>
26. A. G. Snyder, C. D. Johnson, A. E. Gibbs, L. H. Erikson, Nearshore bathymetry data from the Unalakleet River mouth, Alaska, 2019, 2021. <https://doi.org/10.5066/P9238F8K>
27. L. Stals, Efficient solution techniques for a finite element thin plate spline formulation, *J. Sci. Comput.*, **63** (2005), 374–409. <https://doi.org/10.1007/s10915-014-9898-x>
28. L. Stals, S. Roberts, Smoothing large data sets using discrete thin plate splines, *Comput. Visualization Sci.*, **9** (2006), 185–195. <https://doi.org/10.1007/s00791-006-0033-x>
29. H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.*, **4** (1995), 389–396. <https://doi.org/10.1007/BF02123482>
30. M. Wojciech, Kriging method optimization for the process of DTM creation based on huge data sets obtained from MBESs, *Geosciences*, **8** (2018), 433. <https://doi.org/10.3390/geosciences8120433>
31. X. Xu, K. Harada, Automatic surface reconstruction with alpha-shape method, *Visual Comput.*, **19** (2003), 431–443. <https://doi.org/10.1007/s00371-003-0207-1>
32. Z. Zhang, Y. Wang, P. K. Jimack, H. Wang, Meshingnet: A new mesh generation method based on deep learning, In: *Computational Science – ICCS 2020*, Cham: Springer, 2020. [https://doi.org/10.1007/978-3-030-50420-5\\_14](https://doi.org/10.1007/978-3-030-50420-5_14)
33. O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, *The finite element method: Its basis and fundamentals*, Oxford: Elsevier, 2005. <https://doi.org/10.1016/C2009-0-24909-9>

## Supplementary

### A. Triangulation in irregular domain

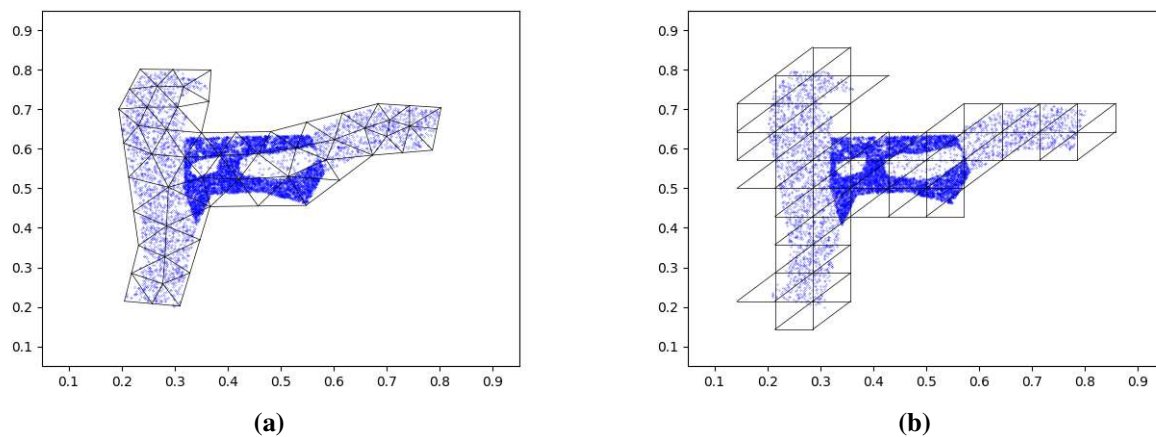
The triangulation in irregular domains is more complicated than in regular rectangular domains and many software packages are available. Ramsay [24] used a PDE toolbox of MATLAB to build arbitrarily shaped triangulated domains. While every data point is a vertex, not every vertex corresponds to a data point. However, this approach only applies to problems with few data points.

Irregular domains of the TPSFEM are built using Python packages `alphashape`<sup>§</sup> and `pygmsh`<sup>¶</sup>. First, a set of vertices are obtained to fit the exterior of the observed data using `alphashape`, which generalizes bounding polygons containing the data [31]. A triangulated mesh is then generated using `pygmsh`. An example FE mesh is shown in Figure A1(a), which fits the observed data closely. Another approach is to remove elements without any data points from FE meshes in square domains as shown in Figure A1(b).

<sup>§</sup>`alphashape`, <https://pypi.org/project/alphashape/>

<sup>¶</sup>`pygmsh`, <https://pypi.org/project/pygmsh/>





**Figure A1.** FE meshes generated by (a) triangulation in an irregular domain; and (b) removing elements from a square domain. The data points are represented as blue dots.

The first approach constructs a bounding polygonal domain from data directly, which adapts to random data distributions and controls the distance from the data to the boundaries. However, it requires parameter tuning, including alpha parameters and mesh sizes, and may not perform well when the data is distributed in a concave polygon. In contrast, the second approach is robust and only requires the mesh size of the initial mesh, which decides the efficiency and distance from the data to the boundaries. It often leads to inconsistent distance from the data to the boundaries, as shown in Figure A1(b). It may also generate meshes with disconnected elements and the initial mesh size should be estimated by the largest distance between nearby data points.

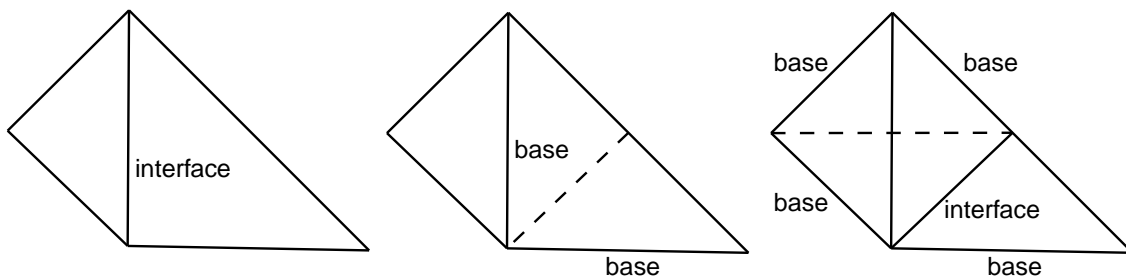
Another major difference between these two approaches is the varied mesh sizes and shapes of the elements. The second approach produces meshes with isosceles right triangular elements with  $45^\circ$  or  $90^\circ$  angles. Using the newest node bisection, the resulting refined meshes will only contain isosceles right triangles. In contrast, the first approach often leads to elements with various sizes and shapes. If angles in initial meshes are bounded between  $[\theta_{min}, \theta_{max}]$ , the angles of the resulting adaptive meshes will be bounded between  $[\frac{1}{2}\theta_{min}, 2\theta_{max}]$  using adaptive refinement with the newest node bisection technique. While they are not as optimal as the second approach, the first approach will not lead to long and thin triangles and the deterioration of interpolation errors.

## B. Adaptive refinement and error indicators

The triangular elements of FE meshes are refined using the newest node bisection method and detailed descriptions are available in [20]. A vertex of each triangle is labeled as the newest node and a triangle is divided into two children triangles by adding a new edge from the newest node to the midpoint of its opposite edge. The midpoint becomes the newest node for further refinement. The angles of the children triangles are bounded, which prevents long, thin triangles and high interpolation errors.

The node-edge data structure is used to store the FE meshes and it is easier to focus on edges for adaptive refinement. In a triangle, the edge that sits opposite to the newest nodes is labeled as a base-edge. Any two triangles that share the same base-edge are called a triangle pair. During uniform refinement, all triangles are bisected along their base-edges and the remaining edges become

base-edges for further refinement. During adaptive refinement, only a subset of the triangle pairs are bisected. Thus, the resulting FE mesh contains triangles of different sizes, which are referred to as fine and coarse triangles. The fine and coarse triangles often share common edges, which are labeled as interface base-edges. If an interface base-edge is to be bisected, the coarse triangle sharing that edge is bisected. The interface base-edge then becomes a base-edge and is bisected as illustrated in Figure B1. This prevents hanging nodes and preserves the compatibility of FE meshes. If the coarse triangle also contains an interface base-edge, a recursive algorithm will be used to iterate through the neighboring coarse triangles until a base-edge is found. This recursive algorithm is guaranteed to terminate.



**Figure B1.** Refinement of interface base-edge.

The error indicators indicate sensitive regions of the domain and mark base-edges with high error indicator values for adaptive refinement. Four error indicators were adapted for the TPSFEM, which are: auxiliary problem, residual-based, recovery-based, and norm-based error indicators [14]. The auxiliary problem error indicator builds a locally more accurate approximation and approximates the error using the difference between it and  $s$ . The residual-based error indicator estimates the residuals of the solution directly. The recovery-based error indicator computes improved gradients of  $s$  and estimates errors by comparing it to  $\nabla s$ . The norm-based error indicator approximates second-order derivatives of smoother  $s$  to indicate large errors. The first two error indicators use local data and the other two do not access data directly. Out of the four error indicators, the auxiliary problem and recovery-based error indicators are compared in the numerical experiments.

The auxiliary problem error indicator's local approximation is obtained by building smoother  $\hat{s} = \hat{\mathbf{b}}(\mathbf{x})^T \hat{\mathbf{c}}$  on a small subset of the domain  $\Omega_e$  using a cluster of neighboring data points  $\{(\hat{\mathbf{x}}_i, \hat{y}_i) : i = 1, 2, \dots, n_e\}$ . It minimizes the functional

$$J_\alpha(\hat{\mathbf{c}}, \hat{\mathbf{g}}_1, \dots, \hat{\mathbf{g}}_d) = \frac{1}{n_e} \sum_{i=1}^{n_e} (\hat{\mathbf{b}}(\hat{\mathbf{x}}_i)^T \hat{\mathbf{c}} - \hat{y}_i)^2 + \alpha \int_{\Omega_e} \sum_{k=1}^d \nabla (\hat{\mathbf{b}}(\hat{\mathbf{x}})^T \hat{\mathbf{g}}_k) \nabla (\hat{\mathbf{b}}(\hat{\mathbf{x}})^T \hat{\mathbf{g}}_k) d\mathbf{x} \quad (\text{B.1})$$

subject to constraint  $\hat{\mathbf{L}}\hat{\mathbf{c}} = \sum_{k=1}^d \hat{G}_k \hat{\mathbf{g}}_k$ . The error norm is then calculated using the difference between  $s$  and  $\hat{s}$ . The error indicator value  $\eta_e$  of an edge  $e$  is defined as

$$\eta_e^2 = \int_{\tau_e} \left[ \sum_{k=1}^d \left( \frac{\partial}{\partial x_k} (s - \hat{s}) \right)^2 \right] d\mathbf{x},$$

where  $\tau_e$  are triangular elements that share edge  $e$ . The accuracy of auxiliary problems depends on several factors, including  $\alpha$ ,  $h$ , size and distribution of data points, and boundary conditions.

The recovery-based error indicator estimates the error by post-processing discontinuous gradients of smoother  $s$  across inter-element boundaries. Since  $s(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}$  consists of piecewise linear basis functions  $\mathbf{b}$ , its gradient approximations  $D^1 s$  are piecewise constants. Improved gradients  $\hat{D}^1 s$  are obtained by  $L^2$ -projection:

$$\int_{\Omega} b_p (\hat{D}^1 s - D^1 s) d\mathbf{x} = 0, \quad p = 1, \dots, m,$$

and the error is estimated as the energy norm of the difference between  $D^1 s$  and  $\hat{D}^1 s$ . Its error indicator value  $\eta_{\tau_e}$  on triangular element  $\tau_e$  is set as

$$\eta_{\tau_e}^2 = \int_{\tau_e} (\hat{D}^1 s - D^1 s)^2 d\mathbf{x}.$$

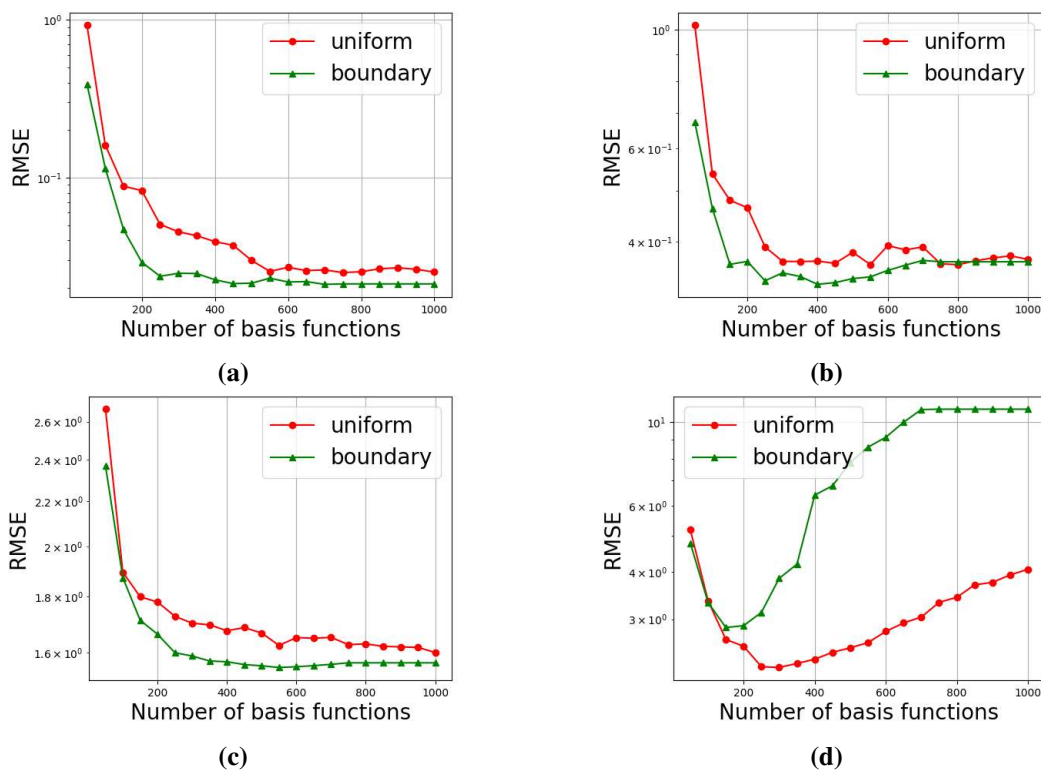
This error indicator only uses information from the FE mesh and is computationally efficient. The descriptions of the newest node bisection and the two error indicators are in two dimensions but they may also be extended to other dimensions.

### C. Accuracy of TPS and its derivatives

A numerical experiment is conducted to evaluate the performance of the TPS for approximating a model problem and its derivatives with the two sampling strategies discussed in Section 4. The test data set is modeled by

$$\begin{aligned} f(x_1, x_2) = & 3(1 - x_1)^2 \exp\left(- (x_1^2) - (x_2 + 1)^2\right) - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) \\ & \exp(-x_1^2 - x_2^2) - \frac{1}{3} \exp\left(- (x_1 + 1)^2 - x_2^2\right) + \varepsilon, \end{aligned} \quad (\text{C.1})$$

where  $\varepsilon$  represents Gaussian noise with mean 0 and standard deviation 0.02. It consists of 10,000 data points uniformly distributed inside the  $[-2.4, 2.4]^2$  region. The TPS interpolant  $t$  is built with  $\hat{n}$  sampled points and  $\alpha$  calculated using the GCV. The quadtree sampling selects data points across the domain and boundary sampling only selects data points outside the  $[-1.9, 1.9]^2$  region. The accuracy of  $t$  and its derivatives is measured using the RMSE, where  $\text{RMSE} = \left(\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (t(\mathbf{x}_i) - f(\mathbf{x}_i))^2\right)^{\frac{1}{2}}$  for  $t$  and  $n_{test}$  is the number of data points outside the  $[-2.2, 2.2]^2$  region. They are chosen to assess the approximations as they better reflect the accuracy near the boundaries. The RMSE for the approximations of the derivatives of  $f$  are defined similarly.



**Figure C1.** RMSE of approximated (a)  $f$ ; (b)  $\frac{\partial f}{\partial x_1}$ ; (c)  $\frac{\partial f}{\partial x_2}$ ; and (d)  $\frac{\partial^2 f}{\partial x_1 x_1} + \frac{\partial^2 f}{\partial x_2 x_2}$  values using TPS kernels.

The convergence of the RMSE for the approximations using the two sampling strategies is shown in Figure C1. As the number of sampled points  $\hat{n}$  increases, the RMSE of all approximations decreases. However, the RMSE for  $f$ ,  $\frac{\partial f}{\partial x_1}$ , and  $\frac{\partial f}{\partial x_2}$  using the boundary sampling strategy stops decreasing after  $\hat{n}$  is larger than 300. For the uniform sampling, the RMSE for  $f$ ,  $\frac{\partial f}{\partial x_2}$ , and  $\frac{\partial f}{\partial x_2}$  is reduced by less than 6%, 1%, and 1%, respectively, after  $\hat{n}$  reaches 350. In contrast, the RMSE for  $\frac{\partial^2 f}{\partial x_1 x_1} + \frac{\partial^2 f}{\partial x_2 x_2}$  starts increasing after  $\hat{n}$  reaches 200 for both sampling strategies, as shown in Figure C1(d). Approximations of second-order derivatives are sensitive to noise and  $\mathbf{w}$  may not be approximated by the second-order derivatives when they are not zero [28].

The boundary sampling achieves lower RMSE for  $f$ ,  $\frac{\partial f}{\partial x_1}$ , and  $\frac{\partial f}{\partial x_2}$  using the same number of points. However, its accuracy for  $\frac{\partial^2 f}{\partial x_1 x_1} + \frac{\partial^2 f}{\partial x_2 x_2}$  deteriorates after  $\hat{n}$  is larger than 200 and has much higher RMSE compared to the uniform sampling. As  $\hat{n}$  continues to increase, the difference in RMSE between the two sampling strategies gradually diminishes.



©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)