Error indicators and adaptive refinement of the discrete thin plate spline smoother

L. Fang[1]

(Received 1 March 2019; revised 29 May 2019)

Abstract

The discrete thin plate spline is a data fitting and smoothing technique for large datasets. Current research only uses uniform grids for this discrete smoother, which may require a fine grid to achieve a certain accuracy. This leads to a large system of equations and high computational costs. Adaptive refinement adapts the precision of the solution to reduce computational costs by refining only in sensitive regions. The error indicator is an essential part of the adaptive refinement as it identifies whether certain regions should be refined. Error indicators are well researched in the finite element method, but they might not work for the discrete smoother as data may be perturbed by noise and not uniformly distributed. Two error indicators are presented: one computes errors by solving an auxiliary problem and the other uses the bounds of the finite element error. Their performances are evaluated and compared with 2D model problems.

DOI:10.21914/anziamj.v60i0.14061 gives this article, © Austral. Mathematical Soc. 2019. Published June 23, 2019, as part of the Proceedings of the 18th Biennial Computational Techniques and Applications Conference . ISSN 1445-8810. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to the DOI for this article.

Contents

1 Introduction

Data fitting and smoothing is an important part of many applications, including image processing and correspondence recovery [2, 6]. To identify the correct trends we need to interpolate the observed data and also smooth noise in the scattered data.

The discrete thin plate spline was developed by Roberts, Hegland and Altas to approximate the thin plate spline by combining the finite element method [4]. Instead of using global basis functions, it utilises piecewise polynomial basis functions with local support. The resulting system of equations is sparse and the size depends only on the number of nodes in the finite element grid, which is more efficient for large datasets.

Current examples of the discrete thin plate spline are produced using uniform grids. The efficiency of the technique can be further improved by using adaptively refined grids to adapt the precision of the solution iteratively within certain sensitive regions. The error indicator is critical in the adaptive refinement process as it evaluates and determines whether to refine a given element. Many error indicators have been developed for the finite element method. However, traditional error indicators might not work for the discrete smoother as the observed data is usually perturbed by noise and not uniformly distributed.

In this article we present two error indicators for the discrete thin plate spline and evaluate their performance through a numerical experiment. Section 2 provides more background information on the thin plate spline and the discrete thin plate spline. Section 3 presents more details of two error indicators for the discrete smoother. Section 4 evaluates the performance of the two error indicators with a pair of model problems.

2 Background

Let $\{(\mathbf{x}_{(i)}, y_{(i)}) : i = 1, 2, \ldots, n\}$ be a set of observed data where $\mathbf{x}_{(i)} \in \mathbb{R}^d$ is a predictor value and $y_{(i)} \in \mathbb{R}$ is a response value. The thin plate spline provides a smooth interpolation $f(\mathbf{x})$ that predicts response values $y = f(\mathbf{x})$ based on given predictor values $\mathbf{x}$. Given n data points, the functional $f(\mathbf{x})$ minimises

$$J_\alpha(f) = \frac{1}{n} \sum_{i=1}^{n} [f(\mathbf{x}_{(i)}) - y_{(i)}]^2 + \alpha \int_\Omega \sum_{|\nu|=2} [D^\nu f(\mathbf{x})]^2 \, d\mathbf{x} \,, \qquad (1)$$

where α is the smoothing parameter, $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)$, $|\nu| = \sum_{j=1}^{d} \nu_j$, and D^ν is a second order vector derivative. The first term of minimisation problem (1) measures fidelity to the data and the second term measures smoothness of the fit. The smoothing parameter α balances the fidelity and the smoothness of the interpolation, which can be calculated automatically using, for example, generalised cross-validation [8]. The resulting system of equations for the minimisation problem (1) is dense and its size depends on the size of the datasets.

The discrete thin plate spline was developed to approximate the thin plate spline and interpolate large datasets efficiently. Instead of using radial basis functions with global support, it uses piecewise polynomial basis functions with local support. This leads to a sparse system of equations whose size depends only on the number of nodes in the finite element grid. The grid can be built independently of the dataset, which is more efficient for fitting large datasets [7].

The discrete smoother $s(\mathbf{x})$ is represented as a linear combination of piecewise linear basis functions $\mathbf{b}(\mathbf{x})$ on $H^1(\Omega)$ finite elements as $s(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\mathbf{c}$, where $\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}), \dots, b_m(\mathbf{x})]^T$, $\mathbf{c} = [c_1, \dots, c_m]^T$ are the corresponding coefficients, m is the number of nodes and Ω is the domain. We choose $H^1(\Omega)$ finite elements because they produce sparse matrices with better conditioning [4]. Since the smoothing term in equation (1) is of order $H^2(\Omega)$ and is not defined for the piecewise linear basis functions $\mathbf{b}(\mathbf{x})$, auxiliary functions $\mathbf{u}$ are introduced to represent the gradient of the discrete smoother s, such that

$$\nabla s = \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1(\mathbf{x})\mathbf{g}_1 \\ \vdots \\ \mathbf{b}_d(\mathbf{x})\mathbf{g}_d \end{bmatrix}, \tag{2}$$

where $\mathbf{g}_1, \dots, \mathbf{g}_d$ are coefficients of the approximation of the gradient. The discrete smoother s and $\mathbf{u}$ satisfy the relationship

$$\int_\Omega \nabla s(\mathbf{x})\nabla b_j(\mathbf{x})\,d\mathbf{x} = \int_\Omega u(\mathbf{x})\nabla b_j(\mathbf{x})\,d\mathbf{x} \tag{3}$$

for every basis function $b_j(\mathbf{x})$. This ensures ∇s and $\mathbf{u}$ are equivalent in a weak sense. Now the order of smoothness is lowered from $H^2(\Omega)$ to $H^1(\Omega)$, and $H^1(\Omega)$ is defined for piecewise linear basis functions of the discrete smoother. The constraint (3) is rewritten as

$$L\mathbf{c} = \sum_{k=1}^{d} G_k\mathbf{g}_k\,, \tag{4}$$

where L is a discrete approximation to the negative Laplacian with $L_{i,j} = \int_\Omega \nabla b_i \cdot \nabla b_j \, d\mathbf{x}$ and G_k is a discrete approximation to the gradient operator with $(G_k)_{i,j} = \int_\Omega b_i \cdot \partial_k b_j \, d\mathbf{x}$.

The minimiser of the discrete smoother is reformulated as

$$J_\alpha(\mathbf{c}, \mathbf{g}_1, \ldots, \mathbf{g}_d) = \mathbf{c}^T A \mathbf{c} - 2\mathbf{d}^T \mathbf{c} + \mathbf{y}^T \mathbf{y}/n + \alpha \sum_{k=1}^{d} \mathbf{g}_k^T L \mathbf{g}_k \,, \tag{5}$$

subject to constraint (4), and where $A = \sum_{i=1}^n \mathbf{b}(\mathbf{x}_{(i)}) \mathbf{b}(\mathbf{x}_{(i)})^T/n$, $\mathbf{y} = [y_{(1)}, \ldots, y_{(n)}]^T$ and $\mathbf{d} = \sum_{i=1}^n \mathbf{b}(\mathbf{x}_{(i)}) y_{(i)}/n$. A single scan of the dataset is required to construct A and $\mathbf{d}$. The minimisation problem (5) is solved using Lagrange multipliers [7].

As the mesh size of the grid decreases, the discrete thin plate spline converges to the model function [4]. In order to obtain an accurate discrete approximation, the grid needs to be resolved with finer elements. Uniform refinement refines all the elements in the domain iteratively, which leads to high computational costs. Adaptive refinement was developed to adapt the precision of a solution dynamically within certain sensitive regions [3]. By only refining part of the domain, the finite element grid can achieve the required accuracy with a smaller system of equations, which leads to fewer computational costs and memory requirements.

3 Error indicators

Error indicators analyse local information and give values to indicate elements with large errors [1, 3]. Many error indicators have been developed to approximate partial differential equations. However, they may not work for the discrete thin plate spline. The discrete smoother uses the scattered data, which is often perturbed by noise and not uniformly distributed. Moreover, the convergence of the discrete smoother depends on $\alpha + d^4 + h^4$, where d is the minimum distance between data points and h is the mesh size [4].

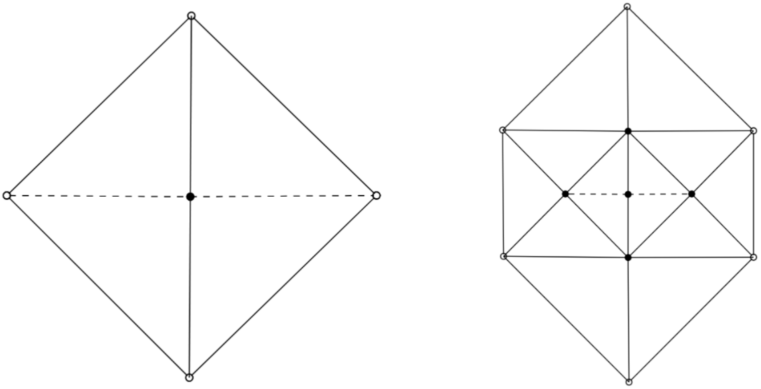
The accuracy of the discrete smoother might not be improved after refining because of factors other than the mesh size h.

Here, two error indicators of the finite element method are modified for the discrete smoother. They are tested using 2D finite element triangular grids and triangles are refined using the newest node bisection [3]. Error indicators discussed in this article focus on determining whether to refine a triangle pair, but the techniques can also be applied to other types of grids.

3.1 Auxiliary problem error indicator

The first error indicator approximates the error by solving an auxiliary problem [3]. The auxiliary problem takes the form of solving a discrete smoother with Dirichlet boundaries to obtain a more accurate solution in the local domain. The local domain can be a triangle pair, as shown in Figure 1(a). The local discrete smoother requires a small system of equations and low computational costs. It can also include more neighbouring triangles, as shown in Figure 1(b), which will increase the accuracy, but also the computational costs. The numerical experiments in this article use the five-point local domain in Figure 1(b). We chose local domain in order to reduce the effects of noise, despite additional computational costs.

The accuracy of the auxiliary problem is improved by refining the triangle pair, which is marked with dashed lines in Figure 1. The improved solution $\hat{s}$ should be a better approximation than the current solution s. The error is approximated by $\|\hat{s} - s\|$, where $\| \cdot \|$ is the energy norm over the original triangle pair. This error indicator is approximating the reduction in error due to refining the triangle pair. Therefore, dividing the triangle pairs with the highest error indicators is equivalent to dividing the triangle pairs that change the solution the most. One limitation of this error indicator is the additional computational costs and memory requirements. Solving auxiliary problems requires the A and d in minimisation problem (5) using data points stored in memory.



(a) one-point domain (b) five-point domain

Figure 1: Two local domains [3].

3.2 Norm-based Error indicator

The second error indicator uses a bound on the L_∞ norm of the error of the finite element solution to calculate indicator values [3, 5]. The accuracy of a piecewise polynomial approximation to the partial differential equation solution using the Galerkin method depends on how accurately the solution can be approximated in the approximating space. Sewell [5] suggested that the error of a 2D finite element solution with piecewise linear basis functions is bounded by $\int_{t_i} D^2_{\max} u \, d\mathbf{x}$, where t_i is ith triangle, u is the model function of a smooth problem and $D^2_{\max} u(x_1, x_2) = \max_{i+j=2} |\partial^2 u(x_1, x_2)/\partial x_1^i x_2^j|$.

In a uniform grid, when the triangles are small enough, $D^2_{\max} u$ is nearly constant in every triangle. Therefore, for a near-optimal grid, $\int_{t_i} D^2_{\max} u \, d\mathbf{x}$ should be approximately equal for all the triangles. This error indicator sets the integral value of a triangle pair as its indicator value. It refines triangles with higher integral values to ensure the integral value is approximately equal for all triangles. It identifies regions where the solution changes rapidly and uses finer elements to achieve the required accuracy in those regions.

Derivative $D^2_{\max}u$ cannot be estimated accurately since u is unknown. Sewell [5] approximted the integral using the current finite element solution s. Since $D^2_{\max}s \equiv 0$ for piecewise linear basis functions, $D^2_{\max}u$ cannot be computed directly using s. Instead, we approximate first order derivatives $D_{x_1}s$ and $D_{x_2}s$ on each triangle, which are gradients of the piecewise flat surface s. The derivative $D_{x_j}s$ of the triangle t_i is denoted as $D_{x_j,t_i}s$. The $D_{x_1}s$ and $D_{x_2}s$ values on each node are estimated as the average of $D_{x_1}s$ and $D_{x_2}s$ values on neigbouring triangles $D_{x_j}s = \sum_{i=1}^{k} D_{x_j,t_i}s/k$, where k is the number of neighbouring triangles. We define new piecewise flat surfaces by using $D_{x_1}s$ or $D_{x_2}s$ on each node as the nodal value. Derivatives $D_{x_1x_1,t_i}s$, $D_{x_1x_2,t_i}s$, $D_{x_2x_1,t_i}s$ and $D_{x_2x_2,t_i}s$ on each triangle t_i are calculated using the new piecewise flat surfaces in the same way as $D_{x_1,t_i}s$ and $D_{x_2,t_i}s$. The derivative of triangle t_i is approximated as

$$D^2_{\max}u(x_1, x_2) \approx \max\{D_{x_1x_1,t_i}s, (D_{x_1x_2,t_i}s + D_{x_2x_1,t_i}s)/2, D_{x_2x_2,t_i}s\} . \qquad (6)$$

4 Model problems

As mentioned in Section 3, the convergence of the discrete smoother depends on the smoothing parameter α, the minimum distance to any data d and mesh size h. The discrete smoother's accuracy cannot be improved by a smaller h if α and d are too large, as shown in Figures 2, 3 and 4 which present three 1D examples. The data points are shown as triangles and the discrete smoother is represented by the blue line. Figure 2 shows a discrete smoother with $\alpha = 10^{-7}$ and $d = 0.01$, which gives a good fit to the model problem. The discrete smoother is paired with a large smoothing parameter $\alpha = 0.0001$ in Figure 3 and in Figure 4 a large d is applied due to missing data. Neither interpolations in Figures 3 and 4 represent the data points as well as the discrete smoother shown in Figure 2.

Figure 5 shows the log-log plots of the three 1D examples in Figures 2, 3 and 4. The L_2 error norm of the discrete smoothers in Figures 3 and 4 are not reduced

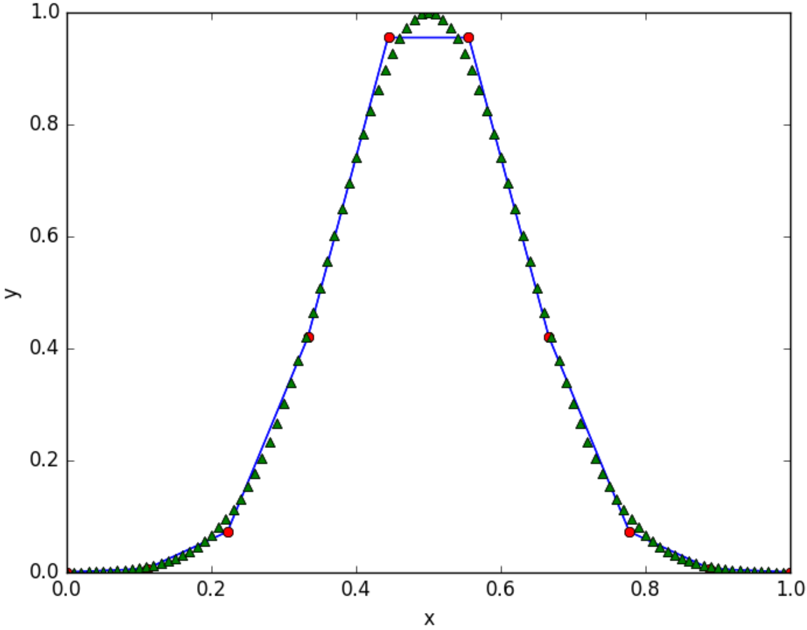


Figure 2: The 1D discrete smoother with standard $\alpha = 10^{-7}$ and $d = 0.01$. Data points $\{(x_{(i)}, y_{(i)})\}$ are green triangles and the discrete smoother $y = f(x)$ is the blue line.

by increasing the number of nodes. The L_2 error norm of the discrete smoother in Figure 2 decreases as the number of nodes are increased in the first three iterations. Then the error halts as h becomes smaller than d, which no longer improves the accuracy of the discrete smoother.

A pair of 2D model problems, shown in Table 1, were derived to test the performance of error indicators. The exponential function has steep gradients, which require finer elements to achieve higher precision. In contrast, the sine function is smoother and is a good test problem to check over-refinement. We tested these model problems with and without noise to verify the error indi-

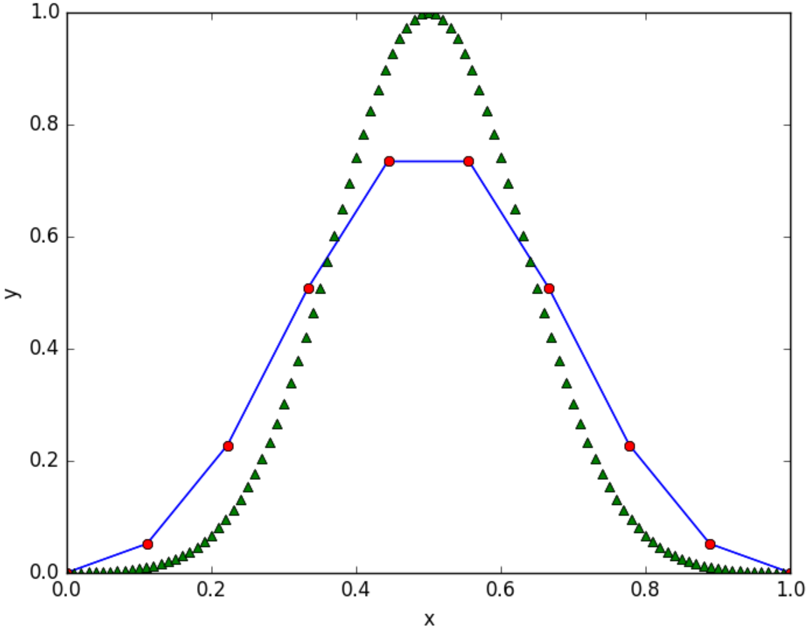


Figure 3: The 1D discrete smoother with $d = 0.01$ and large $\alpha = 0.0001$. Data points $\{(x_{(i)}, y_{(i)})\}$ are green triangles and the discrete smoother $y = f(x)$ is the blue line.

Table 1: Model problems.

Model	$u(\mathbf{x})$	n	α
sine	$\sin(\pi x_1)\sin(\pi x_2)$	250 000	10^{-7}
exponential	$\exp[-30(x_1 - 0.5)^2]\exp[-30(x_2 - 0.5)^2]$	250 000	10^{-7}

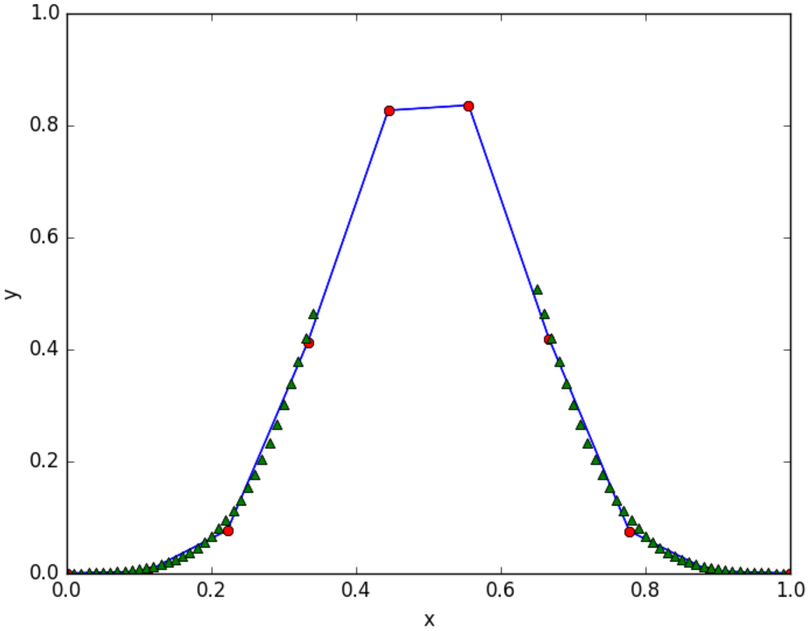


Figure 4: The 1D discrete smoother with $\alpha = 10^{-7}$ and large $d = 0.3$. Data points $\{(x_{(i)}, y_{(i)})\}$ are green triangles and the discrete smoother $y = f(x)$ is the blue line.

cators' performance. The uniform random noise is in the range $[-0.05, 0.05]$. In these test problems, we selected $\alpha = 10^{-7}$, which is an intermediate smoothing parameter for smooth problems. We also chose $d = 0.0015$, which is shorter than the shortest edge in the grid. We choose small values for α and d since we focus on how the error indicators depend on h.

The experiment compares the efficiency of the refined grids, measured by the square root of the number of nodes required to achieve a certain precision. Mesh size h is normally used to indicate the fineness of the grid, but it is not applicable for adaptively refined grids. So the square root of the number of

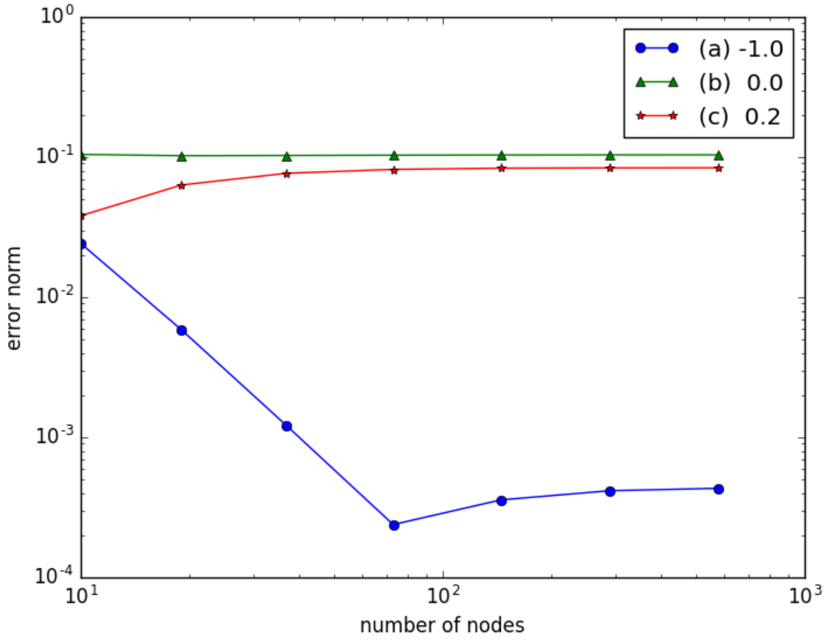


Figure 5: The L_2 error norm dependence on the number of nodes for: (a) $\alpha = 10^{-7}$, $d = 0.01$; (b) $\alpha = 0.0001$, $d = 0.01$; (c) $\alpha = 10^{-7}$, $d = 0.3$. Numbers in the legend are the convergence rates.

nodes is used as it approximates $1/h$ in 2D grids. In each adaptive refinement iteration, error indicators are calculated for each triangle pair and the ones with the highest indicator values were refined until the number of nodes doubled. The error norm is estimated by $\|e\| = \sqrt{\sum_{t_i} h_i^2 e_i^2}$, where h_i is the longest edge in triangle t_i and e_i is the interpolation error at the centre point of triangle t_i. We used the model problem function u as the true solution for datasets without noise as the discrete smoother converges to this true solution. Since the true solution of model problems with noise is unknown, we used a discrete smoother with uniform grids and minimum mesh size $h = 0.005$

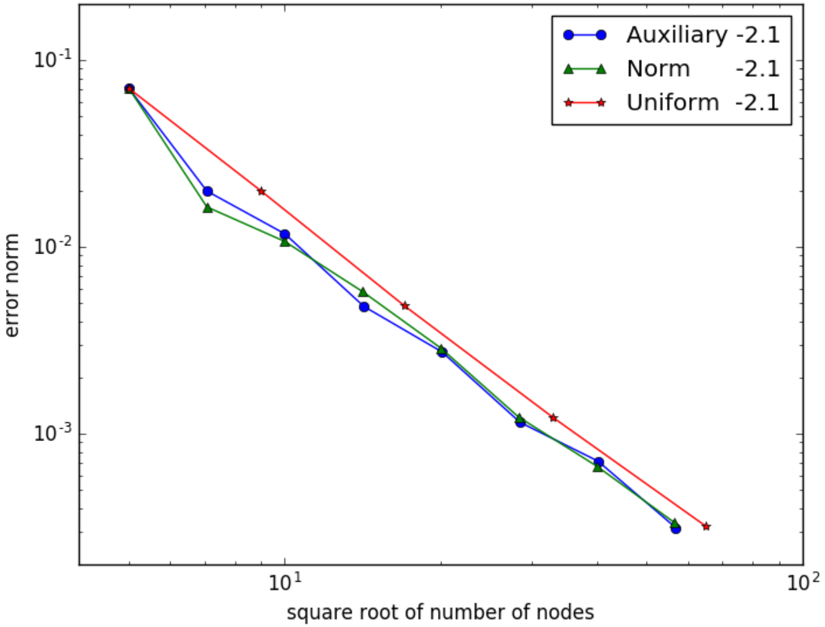


Figure 6: Convergence of the sine model problem $u(\mathbf{x}) = \sin(\pi x_1)\sin(\pi x_2)$ with no noise and using: (i) the auxiliary problem error indicator; (ii) the norm-based error indicator; and (iii) uniform refinement. Numbers in the legend are the convergence rates.

as the fine grid solution to estimate the error norm. We also include the performance of the uniform refinement for comparison.

Figures 6 and 7 are for no noise and uniform noise, respectively, and show log-log plots of the adaptive refinement and the uniform refinement for the sine model. Similarly, Figures 8 and 9 are for no noise and uniform noise, respectively, and show log-log refinement plots for the exponential model.

For the sine model with no noise, Figure 6 shows the adaptive refinement with both error indicators has very close convergence rates, which are slightly

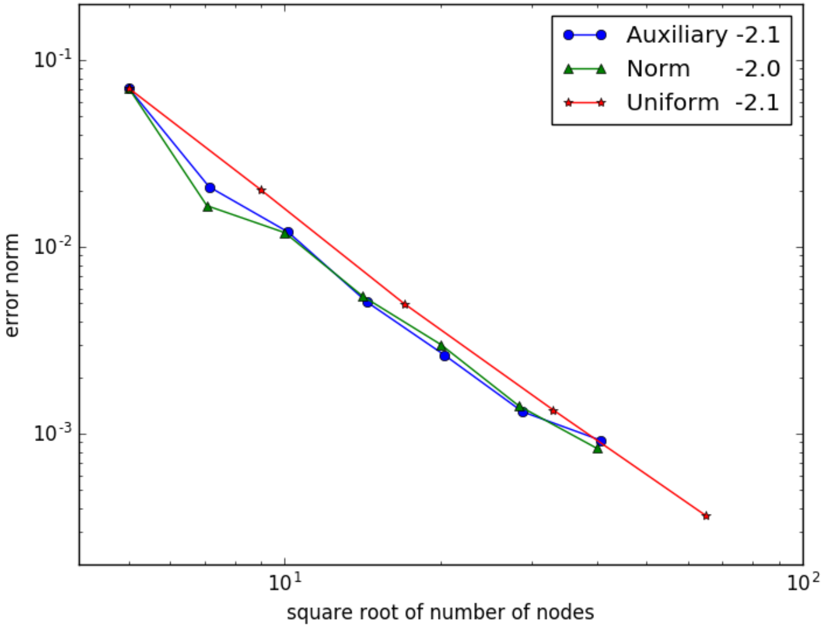


Figure 7: Convergence of the sine model problem $u(\mathbf{x}) = \sin(\pi x_1)\sin(\pi x_2)$ with uniform noise $[-0.05, 0.05]$ and using: (i) the auxiliary problem error indicator; (ii) the norm-based error indicator; and (iii) uniform refinement. Numbers in the legend are the convergence rates.

higher than the uniform refinement. The adaptively refined grids achieve a similar result as the uniform refinement because the sine model problem is smooth. For the exponential model with no noise, Figure 8 shows the adaptive refinement converges much faster than the uniform refinement and both error indicators have similar efficiency and significantly improve the efficiency of the grids. Regions in exponential model with steep gradients are better dealt with by the adaptive refinement, as revealed by relatively lower error norms.

The initial performances of error indicators for both model problems with

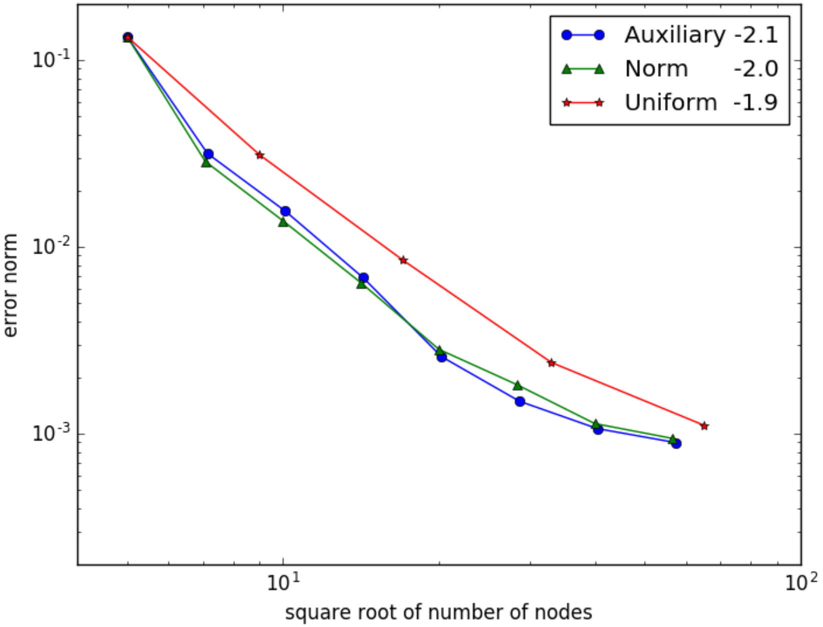


Figure 8: Convergence of the exponential model problem $u(x) = \exp[-30(x_1 - 0.5)^2] \exp[-30(x_2 - 0.5)^2]$ with no noise and using: (i) the auxiliary problem error indicator; (ii) the norm-based error indicator; and (iii) uniform refinement. Numbers in the legend are the convergence rates.

noise are similar to the ones without noise. Both error indicators perform slightly better than the uniform refinement for the sine model problem and improve the efficiency significantly for the exponential model problem, as shown in Figure 7 and 9. However, both performances deteriorate at the last iteration. The error indicators may be misdirected by noise and refine triangles that contribute little to error reduction. Noise in data leads to a discrete smoother with a more oscillatory surface than the original function. The auxiliary problem error indicator utilises data points in the local domain to estimate the error. Noise has more effect on the local approximation for a

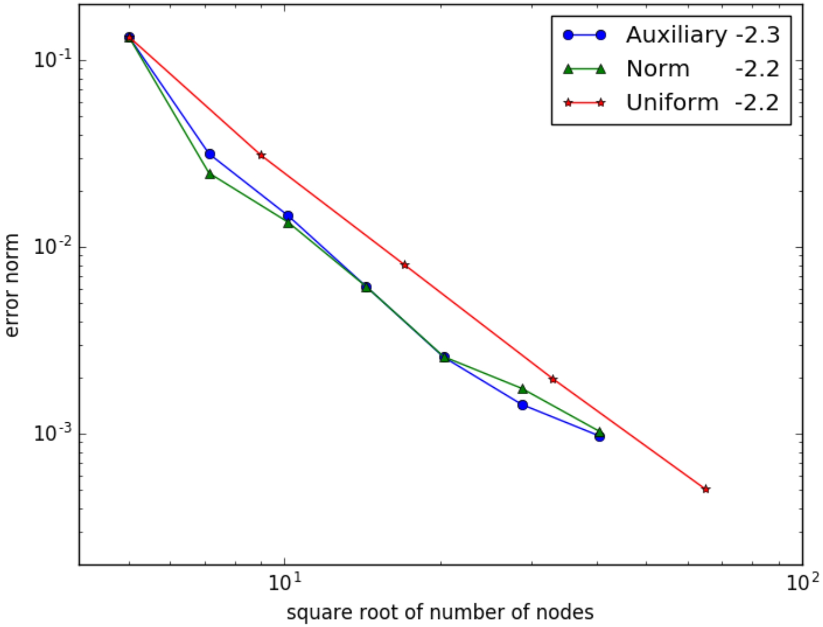


Figure 9: Convergence of the exponential model problem $u(\mathbf{x}) = \exp[-30(x_1 - 0.5)^2]\exp[-30(x_2 - 0.5)^2]$ with uniform noise $[-0.05, 0.05]$ and using: (i) the auxiliary problem error indicator; (ii) the norm-based error indicator; and (iii) uniform refinement. Numbers in the legend are the convergence rates.

smaller local domain with fewer data points. Hence the auxiliary problem error indicator tends to refine oscillatory regions caused by noise instead of the peaks of the model problems. The norm-based error indicator does not use data points and its process of approximating $D^2_{\max}u$ tends to smooth the oscillatory surface caused by noise. The norm-based error indicator only refined the sensitive regions of the original model problems, without reducing the error at the oscillatory regions.

The results of this experiment show that the adaptive refinement improves the efficiency of the discrete smoother, especially for model problems with sensitive regions, such as the exponential function. Both error indicators perform similarly when no noise is present. However, neither error indicator handles noise well, which weakens the efficiency of the resulting grids. The traditional error indicators are not designed to deal with scattered data or noise and they perform less effectively for the discrete smoother. Their performance can be improved by reducing the effects of the noise. For example, $D^2_{\max}u$ approximations of the norm-based error indicator should detect the oscillatory surface that is caused by noise and refine at those regions to reduce the error.

5 Conclusion

In this article, two error indicators of a discrete smoother are presented and evaluated by a numerical experiment on the basis of the solution accuracy per square root of the number of nodes. The error indicators significantly improve the efficiency of the grids for the model problems that have peaks and steep tilts while performing similarly for the smoother model problems. The performance of the error indicators is weakened in the presence of random noise. Traditional error indicators are not as effective for the discrete smoother as for partial differential equations. More work is needed on error indicators and stopping criterion for adaptive refinement of the discrete smoother. They should not only address the noise issue, but also the effects of α and d. I will continue to examine techniques like generalised cross-validation and traditional statistical tests to minimise the effects of noise and test their performance as error indicators.

References

[1] S. C. Brenner and C. Carstensen. *Finite element methods*. Encyclopedia of Computational Mechanics. Wiley, 2017, pp. 1–47. DOI: 10.1002/9781119176817.ecm2003 (cit. on p. C37).

[2] H. Chui and A. Rangarajan. “A new point matching algorithm for non-rigid registration”. In: *Comput. Vis. Image Und.* 89.2–3 (2003), pp. 114–141. DOI: 10.1016/S1077-3142(03)00009-2 (cit. on p. C34).

[3] W. F. Mitchell. “A comparison of adaptive refinement techniques for elliptic problems”. In: *ACM T. Math. Software* 15.4 (1989), pp. 326–347. DOI: 10.1145/76909.76912 (cit. on pp. C37, C38, C39).

[4] S. Roberts, M. Hegland, and I. Altas. “Approximation of a thin plate spline smoother using continuous piecewise polynomial functions”. In: *SIAM J. Numer. Anal.* 41.1 (2003), pp. 208–234. DOI: 10.1137/S0036142901383296 (cit. on pp. C34, C36, C37).

[5] G. Sewell. *Analysis of a finite element method*. Springer-Verlag, 1985. DOI: 10.1007/978-1-4684-6331-6 (cit. on pp. C39, C40).

[6] R. Sprengel, K. Rohr, and H. S. Stiehl. “Thin-plate spline approximation for image registration”. In: *P. IEEE EMBS*. Vol. 3. IEEE. 1996, pp. 1190–1191. DOI: 10.1109/IEMBS.1996.652767 (cit. on p. C34).

[7] L. Stals. “Efficient solution techniques for a finite element thin plate spline formulation”. In: *J. Sci. Comput.* 63.2 (2015), pp. 374–409. DOI: 10.1007/s10915-014-9898-x (cit. on pp. C36, C37).

[8] G. Wahba. *Spline models for observational data*. Vol. 59. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1990. DOI: 10.1137/1.9781611970128 (cit. on p. C35).

Author address

1. **L. Fang**, Mathematical Sciences Institute, Australian National University, ACT 0200, Australia.
mailto:Lishan.Fang@anu.edu.au
orcid:0000-0002-1914-7817