Journal of Computational Physics 229 (2010) 6343–6361

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A high order multivariate approximation scheme for scattered data sets

Qiqi Wang [a,*], Parviz Moin [b], Gianluca Iaccarino [b]

[a] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, United States
[b] Center for Turbulence Research, Stanford University, CA 94305, United States

ARTICLE INFO

Article history:
Received 17 March 2009
Received in revised form 20 April 2010
Accepted 29 April 2010
Available online 16 May 2010

Keywords:
Rational interpolation
Non-linear regression
Function approximation
Approximation order

ABSTRACT

We present a high order multivariate approximation scheme for scattered data sets. Each data point is represented as a Taylor series, and the high order derivatives in the Taylor series are treated as random variables. The approximation coefficients are then chosen to minimize an objective function at each point by solving an equality constrained least squares. The approximation is an interpolation when the data points are given as exact, or a nonlinear regression function when nonzero measurement errors are associated with the data points. Using this formulation, the gradient information on each data point can be used to significantly reduce the approximation error. All parameters of the approximation scheme can be computed automatically from the data points. An uncertainty bound of the approximation function is also produced by the scheme. Numerical experiments demonstrate that although this method is more computationally intensive than traditional methods, it produces more accurate approximation functions.

1. Introduction

Methods for approximating a continuous target function using finite measurements of the function have been extensively studied since ancient times [1]. These methods include interpolation schemes and regression schemes, both have ubiquitous applications in computational science and engineering. In numerical schemes for solving differential equations, approximation schemes are used to reconstruct the continuous solution from its value on a finite set of grid points. Polynomial interpolation is an essential part of spectral methods and finite element methods [2,3]. B-splines [4] and radial basis function approximations [5,6] have also been used in spectral methods and meshless methods for differential equations. In uncertainty quantification, interpolation schemes can be used to construct surrogate response functions of the quantities of interest [7,8]. The statistics of these quantities then can be efficiently sampled from the surrogate functions. In lattice based numerical optimization, approximation schemes are used to construct surrogate functions, based on which the search directions are determined.

Multivariate interpolation was first studied by Borchardt [9] and Kronecker [10]. de Boor and Ron proposed the "least solution" for multivariate polynomial interpolation for non-uniform grid [11]. Multivariate rational interpolation has been studied in [12]. In addition to extensions of one-dimensional interpolation methods, many schemes are proposed specifically for interpolating scattered data sets in multi-dimensional spaces. Shepard's method [13], Kriging [14], and radial basis function interpolation [15,16] are among the most popular multivariate interpolation methods. Comprehensive surveys of scattered data interpolation schemes can be found in [17,18].

We have recently developed a uni-variate interpolation and regression scheme for arbitrary grids [19]. In this scheme, the interpolant function is a rational function with no singularities, and we have proved that the approximation error converges at a rate faster than any polynomial order (i.e., exponential rate).

* Corresponding author.
E-mail addresses: qiqi@mit.edu (Q. Wang), moin@stanford.edu (P. Moin), jops@stanford.edu (G. Iaccarino).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.04.047

In this paper, we extend our uni-variate approximation scheme to multiple dimensions with scattered data points. We also show that "gradient data points" can be used to improve the accuracy of the approximation. In this scheme, high order approximation of the target function on scattered points is achieved by explicitly minimizing the estimated interpolation error at each point. Our method is unique in that we do not specify a class of functions (such as polynomials or linear combination of radial basis functions) in which the approximation function lies. Instead, the approximation function is constructed pointwise by minimizing an estimate of the approximation error at that point. Compared to moving least squares approximation, which is a regression approximation with similar pointwise evaluation, our approximation scheme can either be a interpolation with high rate of convergence, a regression scheme, or a mixed approximation surface.

The remainder of this paper is organized as follows: Section 2 derives the mathematical formulation of the scheme. Section 2.1 defines the mathematical notations used throughout this article, and outlines the mathematical derivation. Sections 2.2 and 2.3 go through the derivation of the scheme step by step. Section 2.4 summarizes the resulting approximation scheme. For readers that are not interested in the mathematical details, it is possible to skip Sections 2.2 and 2.3. In later sections, however, we refer to these sections to explain some characteristics of our approximation scheme.

Section 3 is an essential part of this article. In this section, we characterize our approximation scheme both as an interpolation scheme and as a nonlinear regression scheme, depending on whether the estimated measurement errors associated with the data points are zero. Section 3.1 defines the prediction interval, which is a natural output of our approximation scheme.

Section 4 describes the four parameters that can be controlled by the user. Several examples are given to illustrate how these parameters influence the behavior of the approximation function. Section 5 describes our method of calculating these parameters from data points, which is the internal mechanism of making the "automatic" mode possible.

Section 6.1 provides some examples of our approximation scheme with automatically calculated parameters. Although having a higher computational cost than other methods, it is evident from these examples that our approximation scheme has the advantage of accuracy, robustness, and the ability to use gradient information to improve the approximation. Finally, Section 9 summarizes the topics discussed in this article and outlines further research directions.

2. Mathematical formulation

2.1. Notation and basic formulation of $\tilde{f}$

We consider a target function f in a d-dimensional space. The values of the function is measured at n_v points x_{vi}, $i = 1,\ldots,n_v$. Each x_{vi} is a d-dimensional vector $x_{vi} = (x_{vi1},\ldots,x_{vid})$, representing a point in the d-dimensional space. These points are called "value nodes". The measurement of the function at x_{vi} is $\hat{f}(x_{vi})$, which is $f(x_{vi})$ plus a measurement error. The estimated size of the measurement error $\sigma_{vi} \geqslant 0$ is assumed known. When $\sigma_{vi} = 0$, we assume that the measurement is exact, and $\hat{f}(x_{vi}) = f(x_{vi})$. The data points $(x_{vi}, \hat{f}(x_{vi})),\ i = 1,\ldots,n_v$ are called "value data points".

The gradient of the function is measured at n_g points x_{gi}, $i = 1,\ldots,n_g$. Each x_{gi} is also a d-dimensional vector $x_{gi} = (x_{gi1},\ldots,x_{gid})$, representing a point in the d-dimensional space. These points are called "gradient nodes". The measurement of the gradient of f at x_{gi} is a d-dimensional vector $\widehat{\nabla f}(x_{gi}) = (\widehat{\nabla_1 f}(x_{gi}),\ldots,\widehat{\nabla_d f}(x_{gi}))$, which is the exact gradient $\nabla f(x_{gi})$ plus a measurement error, where the exact gradient

$$\nabla f = (\nabla_1 f(x_{gi}),\ldots,\nabla_d f(x_{gi})), \quad \text{where } \nabla_k f(x_{gi}) = \frac{\partial f}{\partial x_k}(x_{gi}).$$

The estimated size of the measurement error $\sigma_{gi} \geqslant 0$ is again known. When $\sigma_{gi} = 0$, we assume that the measurement of the gradient is exact, and $\widehat{\nabla f}(x_{gi}) = \nabla f(x_{gi})$. The data points $(x_{gi}, \widehat{\nabla f}(x_{gi})),\ i = 1,\ldots,n_g$ are called "gradient data points".

The high order derivatives of f are represented using multi-index notation. Let $\kappa = (\kappa_1,\ldots,\kappa_d)$ be a d-dimensional multi-index, the κ-order derivative of f is defined as

$$f^{(\kappa)} = \frac{\partial^{|\kappa|} f}{\partial x_1^{\kappa_1} \cdots \partial x_d^{\kappa_d}}.$$

The absolute value, also called the total order of κ is defined as

$$|\kappa| = \sum_{k=1}^{d} \kappa_k,$$

and the factorial of κ is defined as

$$\kappa! = \prod_{k=1}^{d} \kappa_k!$$

In addition, the κ order multivariate monomial is defined as

$$x^{\kappa} = x_1^{\kappa_1} \cdots x_d^{\kappa_d}.$$

Specifically, we denote the unit multi-indices as e_k, $k = 1,\ldots,d$, which is defined as

$$e_k = (\delta_{k1},\ldots,\delta_{kd}), \quad \text{where } \delta_{kj} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

These notations are used throughout this paper.

Using the information from the value data points and gradient data points, we construct an approximation function $\tilde{f}$. The specific form of the approximation is

$$\tilde{f}(x) = \sum_{i=1}^{n_v} a_{vi}(x)\hat{f}(x_{vi}) + \sum_{i=1}^{n_g} a_{gi}(x)\cdot\nabla\hat{f}(x_{gi}), \tag{1}$$

where each a_{vi} is a scalar valued function of x, and each a_{gi} is a vector valued function of x. Both a_{vi} and a_{gi} are called "basis functions". When considering a fixed x, we denote the values of the basis functions $a_{vi}(x)$ and $a_{gi}(x)$ as a_{vi} and a_{gi}. We also use the notations $a_v = (a_{v1},\ldots,a_{vn_v})$, $a_g = (a_{g1},\ldots,a_{gn_g})$, and $a = (a_v, a_g)$ to denote the value of all the basis functions at x. Once the basis functions are determined, the approximation function $\tilde{f}(x)$ is determined by Eq. (1).

In the rest of Section 2, we discuss our approach of constructing these basis functions. They are constructed so that the approximation error $\tilde{f}(x) - f(x)$ is small for infinitely differentiable functions. Section 2.3 transforms this objective into a least squares problem, which can be solve to obtain the values of the basis functions.

2.2. Representing the approximation error with Taylor expansions

Incorporating Eq. (1) into the approximation error at some point x in the d-dimensional space, we get

$$\begin{aligned}\tilde{f}(x) - f(x) &= \sum_{i=1}^{n_v} a_{vi}\hat{f}(x_{vi}) + \sum_{i=1}^{n_g} a_{gi}\cdot\nabla\hat{f}(x_{gi}) - f(x) \\ &= \sum_{i=1}^{n_v} a_{vi}f(x_{vi}) + \sum_{i=1}^{n_g} a_{gi}\cdot\nabla f(x_{gi}) - f(x) + \sum_{i=1}^{n_v} a_{vi}(\hat{f}(x_{vi}) - f(x_{vi})) + \sum_{i=1}^{n_g} a_{gi}\cdot(\nabla\hat{f}(x_{gi}) - \nabla f(x_{gi})).\end{aligned} \tag{2}$$

Since we consider the approximation error at a single point x, here we can denote the values of the basis function a_{vi} and a_{gi} at x as a_{vi} and a_{gi} without confusion. This notation will be used throughout the paper.

In order to estimate this approximation error, we expand each $f(x_{vi})$ around x using the multivariate Taylor's theorem:

$$f(x_{vi}) = f(x) + \sum_{0<|\kappa|\leqslant N} \frac{f^{(\kappa)}(x)}{\kappa!}(x_{vi}-x)^{\kappa} + \sum_{|\kappa|=N+1} \frac{f^{(\kappa)}(\xi_{i\kappa})}{\kappa!}(x_{vi}-x)^{\kappa}, \tag{3}$$

where κ is a multi-index notation defined in Section 2.1. The residual term of the Taylor expansion includes the high order derivatives of f at some unknown points $\xi_{i\kappa}$, which is generally different for each i and κ.

The order of the Taylor expansion is chosen so that the number of derivative terms in the expansion matches the degree of freedom provided by the data points. More precisely, N is chosen so that

$$|\{\kappa : |\kappa| < N\}| \approx n_v + dn_g. \tag{4}$$

This choice of Taylor expansion order is determined by the maximum amount of information that can be extracted from the given set of data. Consider a very smooth function whose high order derivatives decay very fast. With n_v value data points and n_g gradient data points, a total of $n_v + dn_g$ number of lowest order function derivatives $f^{(\kappa)}(x)$ can be uniquely determined. This yields an accurate approximation given that higher order derivatives are relatively unimportant. When the function to be approximated is less smooth, data points that are far away is not useful in determining the derivatives of the function. Therefore, the amount of high order information that can be extracted from the data is always bounded by $n_v + dn_g$. For this reason, we always limit the order of our Taylor expansion, so that it contains approximately $n_v + dn_g$ terms.

Similarly, each $\nabla f(x_{gi}) = f^{(e_k)}(x_{gi})$ can also be expanded around x with Taylor's theorem:

$$\begin{aligned}\nabla_k f(x_{gi}) &= \sum_{|\kappa|<N} \frac{f^{(\kappa+e_k)}(x)}{\kappa!}(x_{gi}-x)^{\kappa} + \sum_{|\kappa|=N} \frac{f^{(\kappa+e_k)}(\eta_{ik\kappa})}{\kappa!}(x_{gi}-x)^{\kappa} \\ &= \sum_{\substack{0<|\kappa|\leqslant N \\ \kappa_k>0}} \frac{f^{(\kappa)}(x)}{(\kappa-e_k)!}(x_{gi}-x)^{\kappa-e_k} + \sum_{\substack{|\kappa|=N+1 \\ \kappa_k>0}} \frac{f^{(\kappa)}(\eta_{ik\kappa})}{(\kappa-e_k)!}(x_{gi}-x)^{\kappa-e_k},\end{aligned} \tag{5}$$

where e_k is the unit multi-index defined in Section 2.1. The residual term of the Taylor expansion includes the high order derivatives of f at some unknown points $\eta_{ik\kappa}$, which is generally different for each i, k and κ.

Incorporating these Taylor expansions into Eq. (2), the error of the approximation can be represented as

$$\tilde{f}(x)-f(x)=\sum_{|\kappa|\leqslant N} f^{(\kappa)}(x)\mathcal{X}_\kappa(x;a)+\sum_{|\kappa|=N+1}\sum_{i=1}^{n_v} f^{(\kappa)}(\xi_{i\kappa})\left(a_{vi}\frac{(x_{vi}-x)^\kappa}{\kappa!}\right)+\sum_{|\kappa|=N+1}\sum_{\substack{k=1\\ \kappa_k>0}}^{d}\sum_{i=1}^{n_g} f^{(\kappa)}(\eta_{ik\kappa})\left(a_{gik}\frac{(x_{gi}-x)^{\kappa-e_k}}{(\kappa-e_k)!}\right)$$
$$+\sum_{i=1}^{n_v}(\hat{f}(x_{vi})-f(x_{vi}))a_{vi}+\sum_{k=1}^{d}\sum_{i=1}^{n_g}(\nabla_k\hat{f}(x_{gi})-\nabla_k f(x_{gi}))a_{gik}, \tag{6}$$

where

$$\mathcal{X}_\kappa(x;a)=\begin{cases}\sum_{i=1}^{n_v} a_{vi}-1, & |\kappa|=0,\\ \sum_{i=1}^{n_v} a_{vi}\frac{(x_{vi}-x)^\kappa}{\kappa!}+\sum_{\substack{k=1\\ \kappa_k>0}}^{d}\sum_{i=1}^{n_v} a_{gik}\frac{(x_{gi}-x)^{\kappa-e_k}}{(\kappa-e_k)!}, & |\kappa|>0.\end{cases} \tag{7}$$

Eq. (6) splits the approximation error $\tilde{f}(x)-f(x)$ into three distinct parts:

1. The first line represents the contributions from the derivatives of the f at x, i.e.,
$$f^{(\kappa)}(x),\quad 0\leqslant|\kappa|\leqslant N.$$
These quantities are unknown; however, their coefficients in Eq. (6), $\mathcal{X}_\kappa(x;a)$, are functions of x and the values of the basis functions a_{vi} and a_{gi} at x.
2. The second and third lines are the residuals of the Taylor expansions. They represent the contribution from the high order derivatives of f at unknown points $\xi_{i\kappa}$ and $\eta_{ik\kappa}$, i.e.,
$$f^{(\kappa)}(\xi_{i\kappa}),\ i=1,\dots,n_v\quad\text{and}\quad f^{(\kappa)}(\eta_{ik\kappa}),\ i=1,\dots,n_g,\ k=1,\dots,d$$
for all $|\kappa|=N+1$. These quantities are unknown; however, their coefficients in Eq. (6),
$$\left(a_{vi}\frac{(x_{vi}-x)^\kappa}{\kappa!}\right)\quad\text{and}\quad\left(a_{gik}\frac{(x_{gi}-x)^{\kappa-e_k}}{(\kappa-e_k)!}\right),$$
are functions of x and the values of the basis functions a_{vi} and a_{gi} at x.
3. The last line represents the contribution from the measurement errors
$$\hat{f}(x_{vi})-f(x_{vi})\quad\text{and}\quad\nabla_k\hat{f}(x_{gi})-\nabla_k f(x_{gi}).$$
These measurement errors are unknown; however, their coefficients in Eq. (6) are simply the values of the basis functions a_{vi} and a_{gi} at x. When measurement errors are absent, this part is absent from Eq. (6).

2.3. Least squares for the approximation error

We constrain the basis functions with

$$\sum_{i=1}^{n_v} a_{vi}\equiv 1, \tag{8}$$

so that the approximation (1) is exact for a constant function f. With this constraint, the formula for the approximation error (6) becomes

$$\tilde{f}(x)-f(x)=\sum_{0<|\kappa|\leqslant N} f^{(\kappa)}(x)\mathcal{X}_\kappa(x;a)+\sum_{|\kappa|=N+1}\sum_{i=1}^{n_v} f^{(\kappa)}(\xi_{i\kappa})\left(a_{vi}\frac{(x_{vi}-x)^\kappa}{\kappa!}\right)$$
$$+\sum_{|\kappa|=N+1}\sum_{\substack{k=1\\ \kappa_k>0}}^{d}\sum_{i=1}^{n_g} f^{(\kappa)}(\eta_{ik\kappa})\left(a_{gik}\frac{(x_{gi}-x)^{\kappa-e_k}}{(\kappa-e_k)!}\right)+\sum_{i=1}^{n_v}(\hat{f}(x_{vi})-f(x_{vi}))a_{vi}+\sum_{k=1}^{d}\sum_{i=1}^{n_g}(\nabla_k\hat{f}(x_{gi})-\nabla_k f(x_{gi}))a_{gik}. \tag{9}$$

This equation represents the approximation error at each point as a linear combination of the following unknowns: the derivatives $f^{(\kappa)}(x)$, $f^{(\kappa)}(\xi_{i\kappa})$, $f^{(\kappa)}(\eta_{ik\kappa})$, and the measurement errors $\hat{f}(x_{vi})-f(x_{vi})$ and $\nabla_k\hat{f}(x_{gi})-\nabla_k f(x_{gi})$. We construct the following weighted L_2 norm of this linear combination:

$$\mathcal{Q}(x;a)=\sum_{0<|\kappa|\leqslant N} w_{|\kappa|}^2\mathcal{X}_\kappa^2(x;a)+\sum_{|\kappa|=N+1}\sum_{i=1}^{n_v} w_{|\kappa|}^2\left(a_{vi}\frac{(x_{vi}-x)^\kappa}{\kappa!}\right)^2+\sum_{|\kappa|=N+1}\sum_{\substack{k=1\\ \kappa_k>0}}^{d}\sum_{i=1}^{n_g} w_{|\kappa|}^2\left(a_{gik}\frac{(x_{gi}-x)^{\kappa-e_k}}{(\kappa-e_k)!}\right)^2$$
$$+\sum_{i=1}^{n_v}\sigma_{vi}^2a_{vi}^2+\sum_{k=1}^{d}\sum_{i=1}^{n_g}\sigma_{gi}^2a_{gik}^2, \tag{10}$$

where the weights are

$$w_j = \beta\gamma^j, \quad j = 0, \ldots, N+1. \tag{11}$$

The purpose of the weighted L_2 norm $\mathcal{Q}(x;a)$ defined in Eq. (10) is to reflect the size of the approximation error in Eq. (9). The coefficients of the κ order derivatives in Eq. (9) are weighted by $w_{|\kappa|}$ in Eq. (10); and the coefficients of the measurement errors in Eq. (9) are weighted by their estimated sizes σ_{vi} and σ_{gi} in Eq. (10). The parameters "magnitude" β and "wavenumber" γ control the weights w_j. They are two important parameters of our scheme. For $\mathcal{Q}(\S)$ to best reflect the size of $\tilde{f}(x) - f(x)$, these should be chosen so that the weights $w_j = \beta\gamma^j$ reflect the size of $f^{(\kappa)}(x)$, $|\kappa| = j$. Sections 4.2 and 4.1 discuss how β and γ affect the behavior of the approximation $\tilde{f}(x)$. Section 5 shows our method of automatically calculating these parameters from the given data points.

The values of the basis functions at x is calculated by solving an equality constrained least squares problem

$$\min \mathcal{Q}(x;a) \quad \text{s.t.} \quad \sum_{i=1}^{n_v} a_{vi} = 1, \tag{12}$$

where $\mathcal{Q}(x;a)$ is defined in Eq. (10). The value of the approximation $\tilde{f}(x)$ is then determined by Eq. (1).

2.4. The approximation scheme

The following scheme summaries the construction of the approximation function $\tilde{f}(x)$:

1. Gather the input data: the value data points $(x_{vi}, \hat{f}(x_{vi}))$, the gradient data points $(x_{gi}, \nabla\hat{f}(x_{gi}))$, and the estimate of their measurement errors σ_{vi} and σ_{gi}.
2. Determine the two parameters of the scheme[1]: the "wavenumber" γ and the "magnitude" β.
3. For each point x where $\tilde{f}(x)$ is desired, construct the matrices for the constraints and the least squares. The elements of these matrices are simply the coefficients of a_{vi} and a_{gik} in Eqs. (7) and (10).
4. For each point x where $\tilde{f}(x)$ is desired, solve the equality constrained least squares (12) for $a_{vi}(x)$ and $a_{gik}(x)$.[2]
5. Calculate $\tilde{f}(x)$ for each x using Eq. (1).

3. $\tilde{f}$ as an interpolation and regression approximation

In this section, we first show that the approximation $\tilde{f}$ is an interpolation approximation when the measurement errors of the value data points are zero. In fact, when $\sigma_{vi} = 0$ and $x = x_{vi}$, the solution for the constraint least squares (12) is

$$\begin{aligned} a_{vi'} &= \begin{cases} 1, & i' = i, \\ 0, & i' \neq i, \end{cases} \\ a_{gi'} &= 0, \quad \forall i'. \end{aligned} \tag{13}$$

This can be verified by the following: Incorporating (13) and $x = x_{vi}$ into Eqs. (7) and (10), we get $\mathcal{X}_\kappa(x;a) = 0\ \forall\kappa$ and $\mathcal{Q}(x;a) = 0$. Therefore, the constraint (8) are satisfied while $\mathcal{Q}(x;a)$ is minimized. Thus, (13) is the solution of (12). Now we incorporate (13) into Eq. (1), we get

$$\tilde{f}(x) = \hat{f}(x_{vi}). \tag{14}$$

In other words, the approximation function $\tilde{f}(x)$ goes through the value data point $(x_{vi}, \hat{f}(x_{vi}))$. Therefore, our scheme of constructing $\tilde{f}(x)$ as an interpolation scheme when $\sigma_{vi} = 0$.

When the measurement errors are nonzero, $\tilde{f}$ behaves like a regression function, which does not go through each data point. Fig. 1 shows the approximation function $\tilde{f}(x)$ for the same value data points with different σ_{vi}. As can be seen, when the $\sigma_{vi} = 0$, $\tilde{f}(x)$ is an interpolation function. As σ_{vi} increases, $\tilde{f}(x)$ becomes less oscillatory, and behaves more like a regression function. When σvi have different values at different data points, $\tilde{f}(x)$ interpolates through the data points whose $\sigma_{vi} = 0$, and is closer to the data points whose σ_{vi} are small.

3.1. Prediction interval

The prediction interval of our approximation scheme is constructed as following:

$$(\tilde{f}(x) - c\sigma(x), \tilde{f}(x) + c\sigma(x)), \quad \text{where } \sigma(x) = \sqrt{\mathcal{Q}(x;a)}. \tag{15}$$

[1] These parameters can be set to default values or automatically calculated without significantly affecting the performance of the scheme. See Section 5 for details.

[2] To solve this equality constrained least squares, We use the algorithm described in Section 7 of [19] when $P = 1$, and Algorithm 12.1.2 in [20] for other values of P.

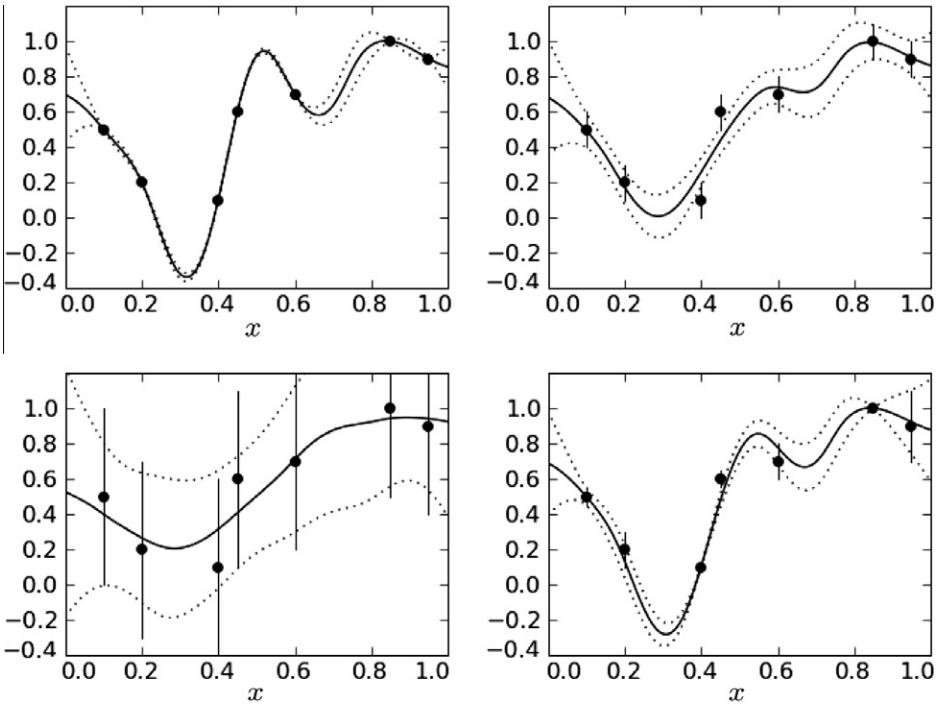

Fig. 1. $\tilde{f}$ calculated on the same data points with different σ_{vi}. $\beta = 0.5$, $\gamma = 10$. Filled circles represent the data points; the vertical bars indicates the size of σ_{vi}; the solid line is $\tilde{f}(x)$; the dotted lines represent the prediction interval $(\tilde{f}(x) - \sigma(x), \tilde{f}(x) + \sigma(x))$. $\sigma_{vi} = 0$ in the upper-left plot; $\sigma_{vi} = 0.1$ in the upper-right plot; $\sigma_{vi} = 0.5$ in the lower-left plot; in the lower-right plot, σ_{vi} has different values at each data point.

The factor c is usually chosen between 1 and 3. A larger c produces a more conservative prediction interval.

In Fig. 1, the confidence interval with $c = 1$ is plotted as dotted lines. As can be seen, when the measurement errors are 0, the size of the prediction interval is zero at the data points. The size of the prediction interval increases as x moves away from the data points. In addition, the larger the size of the measurement errors σ_{vi}, the larger the prediction interval is. Finally, we observe that the prediction intervals are generally larger in the extrapolation region than in the interpolation region.

This prediction interval can be a powerful tool of quantifying uncertainties generated by interpolating between data points. In the case of regression, where the uncertainties of data points are given as σ_{vi}, the prediction interval can be used to propagate the uncertainties from the input data points to the interpolated values. In addition, the prediction interval can be used to automatically estimate the wavenumber parameter γ, as discussed in detail in Section 5.

4. Parameters of the approximation scheme

4.1. The wavenumber γ

The wavenumber γ determines the rate of growth or rate of decay of the weights w_j in Eq. (10). When γ is larger than 1, w_j grows as j increases. In this case, heavier weights are assigned to the higher order derivatives in constructing the objective function $\mathcal{Q}$ defined in Eq. (10). Thus, when we substitute the solution of the least squares into Eq. (9), the contribution to the approximation error $\tilde{f}(x) - f(x)$ from the high order derivatives is smaller compared to the contribution to the approximation error from the lower order derivatives. As a result, as γ increases, the approximation function $\tilde{f}(x)$ visually appears to be more like a lower order approximation. On the other hand, when γ is smaller than 1, w_j decays as j increases. In this case, heavier weights are assigned to the lower order derivatives in constructing the objective function $\mathcal{Q}$. Thus, when we plug the solution of the least squares into Eq. (9), the contribution to the approximation error from the high order derivatives is larger compared to the contribution to the approximation error from the lower order derivatives. As a result, as γ decreases, the approximation function $\tilde{f}(x)$ visually appears to be more like a higher order approximation.

Fig. 2 shows the approximation function $\tilde{f}(x)$ for the same set of data points with four different wavenumber parameter γ. The data points indicated by the black circles are exact measurements of the target function $f(x) = e^{-4(x_1 - x_2)^2}$. σ_{vi} are set to zero, and the approximation function $\tilde{f}(x)$ is a two-dimensional interpolation surface. As can be seen, when γ is small, the approximation surface appears to be a smooth function full of tension. Larger overshoots occur in regions where the data points are sparse. Conversely, when γ is large, the approximation function looks more like a piecewise constant approximation. Steep slopes divide flat plateaus encompassing each data point. In contrast to the small γ case, the approximation function never overshoot the data points.

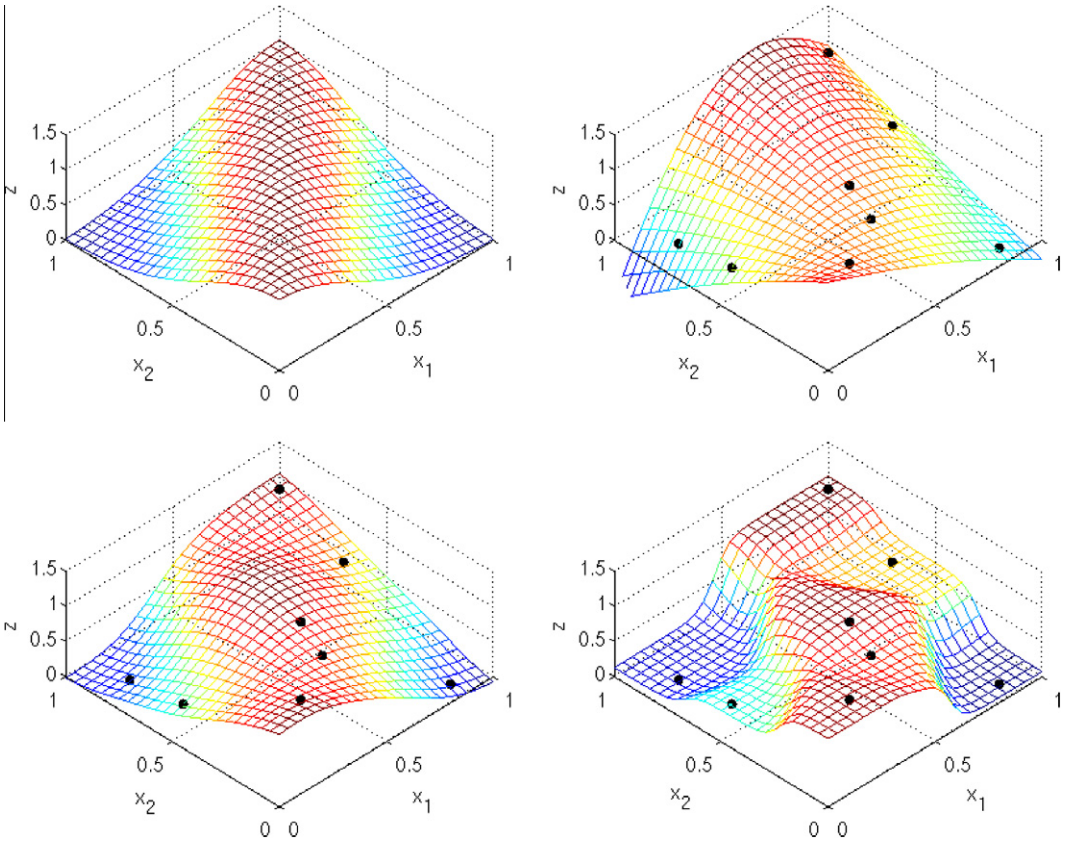

Fig. 2. The target function $f(x) = e^{-4(x_1-x_2)^2}$ (upper-left); and the approximation function $\tilde{f}(x)$ constructed on the same data points with different wavenumber γ: $\gamma = 0.5$ (upper-right), $\gamma = 3.0$ (lower-left), $\gamma = 10.0$ (lower-right). The magnitude parameter $\beta = 0.5$. Both the z-axis and the color indicate the function value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 3 further illustrates the effect of γ by plotting the basis function $a_{vi}(x)$, where i corresponds to the data point near the center of the domain. Again, we see large overshoot in the basis function with the smaller γ, and only slight overshoot in the large γ case. In addition to the different magnitudes of overshoot, we observe that the supports of the basis functions are different in the two cases. When the wavenumber γ is large, the basis function has a local support, i.e., it is nonzero only in a small region around the x_{vi}, and is almost zero elsewhere. As a result, the data point at x_{vi} has significant effect on the value of the approximation function $\tilde{f}(x)$ only in the small region around x_{vi}. From another perspective, the value of the approximation function at each point is determined by the value of its nearby one or two data points. As γ decreases,

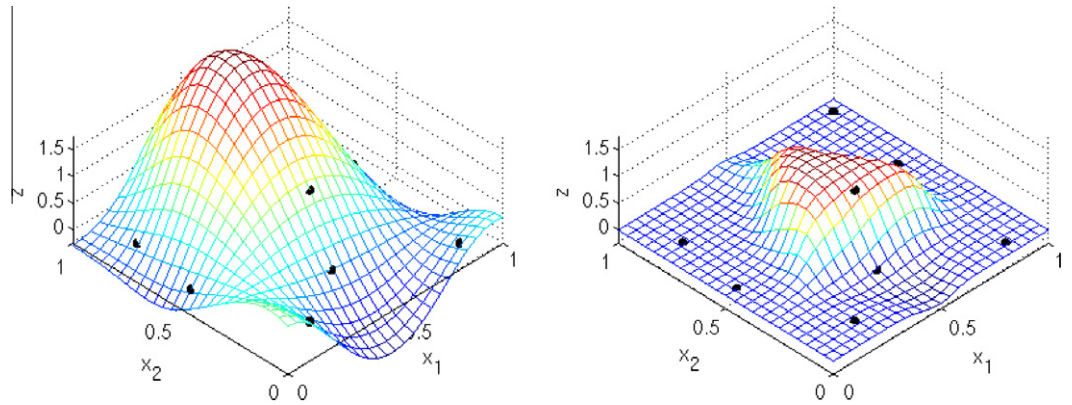

Fig. 3. The basis function $a_{vi}(x)$ on the same grid with different wavenumber γ: $\gamma = 2.0$ (left), $\gamma = 10.0$ (right). The magnitude parameter $\beta = 0.5$. Both the z-axis and the color indicate the value of the basis functions.

the support of the basis functions increases. As a result, the value of the approximation function $\tilde{f}(x)$ at each point is affected by additional surrounding data points. When γ is very small, the basis function is nonzero everywhere, and the approximation function at every x is determined by the collective effect of all the data points. In conclusion, $\tilde{f}$ gradually changes from a local approximation to a global approximation as γ decreases.

The upper plot in Fig. 4 shows the L_2 and L_∞ norms of the approximation error, the difference between the approximation function and the target function $f(x) = e^{-4(x_1-x_2)^2}$. As can be seen, the approximation deteriorates when γ is either too small or too large. When γ is large, the support of the basis functions are too small. As a result, $\tilde{f}(x)$ is constructed too conservatively, based on the information from only one or two data points near x. In this case, decreasing γ allows using more surrounding data points to construct the approximation at each point, increasing its accuracy. On the other hand, however, a too small γ is much more dangerous than a too large γ. The overshoots in regions where data points are relatively sparse can cause the approximation error to be very large. When the number of data points is large, the overshoots and oscillations caused by a too small γ can be orders of magnitude larger than the target function itself. Therefore, choosing a value of γ that is neither too large nor too small is critical for $\tilde{f}(x)$ to be an accurate approximation.

The question of what value of γ is optimal can be answered by analyzing the lower plot in Fig. 4. This plot shows the L_2 and L_∞ norms of the prediction interval $\sigma(x)$, as defined in Eq. (15), as a monotone increasing function of γ. Comparing it with the upper plot, we observe that the approximation error reaches minimum when it has comparable size as the prediction interval. This is not a coincidence. As discussed in Section 2.3, the best choice of γ is when the weights $w_j = \beta\gamma^j$ correctly reflect the growth rate of the jth total order derivatives. With this γ, the objective function $\mathcal{Q}$ defined in Eq. (10) reflects the squared size of the approximation error as shown in Eq. (9). Consequently, the size of the prediction interval $\sigma(x)$, defined as the square root of $\mathcal{Q}$, has the same size as the approximation error. On the other hand, our objective of solving the least squares (12) to minimize $\mathcal{Q}$ is to make the approximation error small. This objective is best achieved when the value of γ is chosen so that the size of $\mathcal{Q}$ reflects the squared approximation error. Therefore, when the value of γ is chosen so that the prediction interval has the same size as the approximation error, the approximation error is minimized. This principle is used to determine the best value of γ from the data point in Section 5.

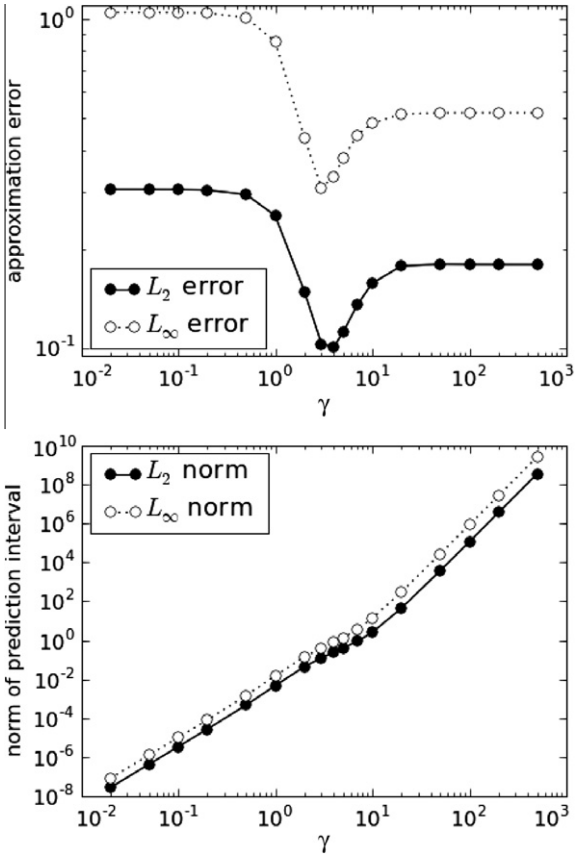

Fig. 4. Norms of the approximation error $\tilde{f}(x) - f(x)$ (upper plot) and the prediction interval $\sigma(x)$ (lower plot) plotted against the wavenumber γ. The target function is $f(x) = e^{-4(x_1-x_2)^2}$. The magnitude parameter $\beta = 0.5$.

4.2. The magnitude β

The magnitude β controls the size of all the weights w_j in Eq. (10). It affects $\tilde{f}(x)$ only when the estimated measurement errors σ_{vi} and σ_{gi} are nonzero. When the measurement errors are absent, β^2 acts as a constant multiplier in constructing the objective function $\mathcal{Q}$. Therefore, it has no effect on the solution a of the least squares (12) and the approximation function $\tilde{f}$. But when the measurement error terms in Eq. (10) are nonzero, β determines the weight of the measurement error term relative to all other terms.

Fig. 5 illustrates the effect of β on $\tilde{f}(x)$ by plotting the approximation surface $\tilde{f}(x)$ with different values of β constructed on the same set of data points. The 50 data points are indicated by the black circles, and the distance from the approximation surface to the data points are indicated by the vertical lines. The location of these data points are generated randomly. Their values equal to the target function $f(x) = x_1^2 + x_2^2$ plus simulated measurement errors, which is independent Gaussian pseudo random numbers with mean 0 and standard deviation 0.1. The approximation functions are constructed with $\sigma_{vi} = 0.1$.

As can be seen in Fig. 5, β determines the sensitivity of the approximation function to the value of each data point. When β is small, the approximation surface is not sensitive to any single data point. It visually looks more like a nonlinear regression surface, and is far away from many data points. In contrast, when β is large, the approximation surface is sensitive to each data point. It visually looks more like a interpolation surface. The surface is much more oscillatory, and almost goes through all the data points.

In fact, increasing β is equivalent of decreasing all the estimated measurement errors. The reason for this effect can be found in Eq. (10). If the value of β and all σ_{vi} and σ_{gi} are doubled at the same time, the value of $\mathcal{Q}(x)$ is simply multiplied by a factor of 4, and the solution of the least squares (12) remains the same. Therefore, the same approximation is obtained should one doubles the value of β, or half all values of σ_{vi} and σ_{gi}.

Fig. 6 plots the error norm of the approximation with respect to β. It shows that the approximation is optimal when β is neither too small nor too large. When β is small, the approximation function is too far away from many data points. As can be seen in the upper-left plot of Fig. 5, the approximation surface is too high in the center and too low near the edges compared to the target function $f(x) = x_1^2 + x_2^2$. The approximation error would have been reduced if the surface could bend more and fit

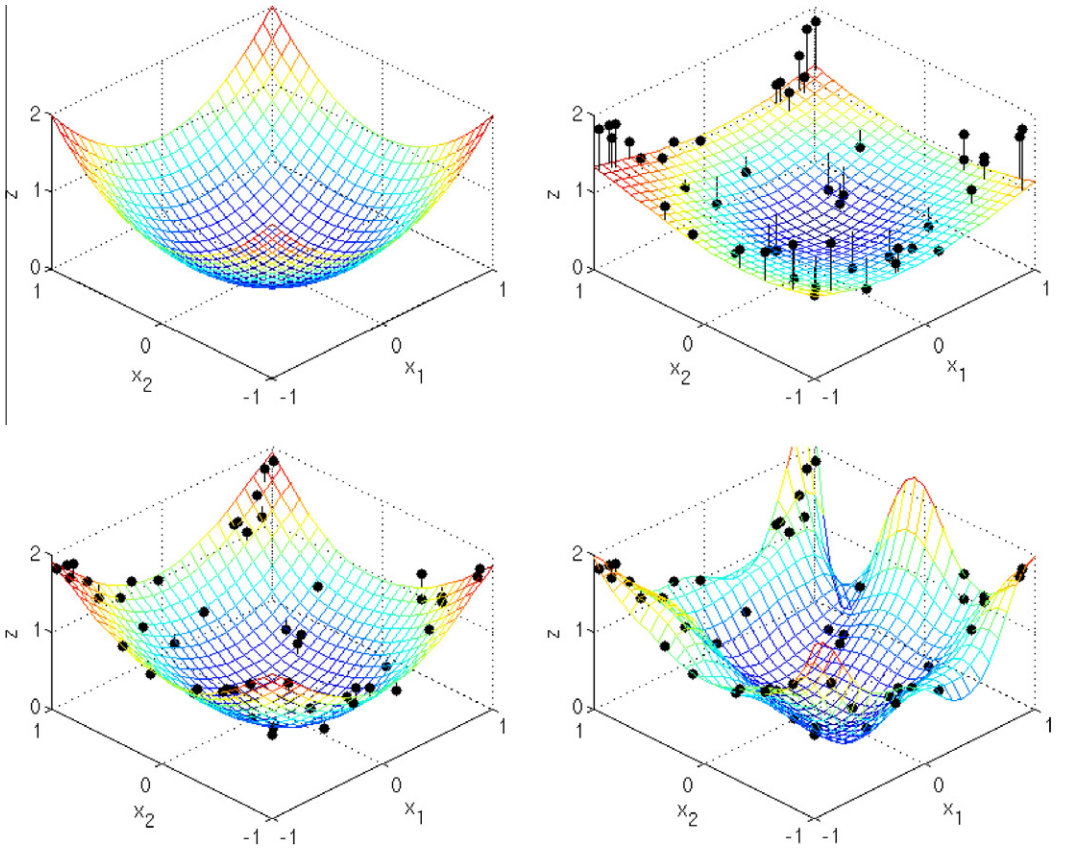

Fig. 5. The target function $f(x) = x_1^2 + x_2^2$ (upper-left); and approximation function $\tilde{f}(x)$ constructed on the same data points with different magnitude β: $\beta = 0.02$, $\beta = 0.5$ (lower-left), $\beta = 100.0$ (lower-right). The data points are corrupted with Gaussian random numbers to simulate measurement errors. The wavenumber parameter $\gamma = 2.0$. Both the z-axis and the color indicate the function value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

closer to the data points. On the other hand, when β is large, the approximation function is too close to the data points that are corrupted by large measurement errors, causing it to be oscillatory. As can be seen in the lower-right plot of Fig. 5, the approximation error would have been reduced if the surface could be smoother by staying away from several data points with obviously large measurement errors. Therefore, choosing a value of β that is neither too large nor too small is important for $\tilde{f}(x)$ to be an accurate approximation in the presence of measurement errors.

The optimal value of β can be found in the definition of the weights $w_j = \beta\gamma^j$. When $j = 0$, we have $\beta = w_0$. Since the best choice of w_j should reflect the size of the jth total order derivatives of f, the best choice of β should reflect the magnitude of variation of f. In our example of $f(x) = x_1^2 + x_2^2$ in the box $-1 < x_1 < 1$, $-1 < x_2 < 1$, the magnitude of variation of the function from its mean is about 1. Consequently, the optimal value of β is around 1. In Section 5, we use this rule to determine the best value of β from the data points.

5. Determination of the parameters from data points

This section proposes a method for automatically determining a suitable set of parameters γ and β from any given data points. We provide some argument on why we use these formulas to automatically calculate the parameters. Although the parameters calculated using this method are rarely the optimal set of values, they have been demonstrated to produce satisfactory approximation functions in practice.

5.1. The magnitude β

As discussed in Section 4.2, the optimal value of β should reflect the magnitude of variation of the target function $f(x)$, which can be estimated by the standard deviation of the data points, i.e.

$$\beta = \sqrt{\frac{\sum_{i=1}^{n}(f(x_{vi}) - \bar{f})^2}{n-1}}, \quad \text{where } \bar{f} = \frac{\sum_{i=1}^{n} f(x_{vi})}{n}.$$

In the presence of measurement error, $f(x_{vi})$ are unknown, so we use the formula

$$\beta = \sqrt{\frac{\sum_{i=1}^{n}(\hat{f}(x_{vi}) - \bar{\hat{f}})^2}{n-1}} \tag{16}$$

to estimate the parameter.

5.2. The wavenumber γ

As discussed in Section 4.1, the optimal value of γ is achieved when the prediction interval $\sigma(x)$ has the same magnitude as the approximation error $\tilde{f}(x) - f(x)$. This rule allows us to determine γ from the given data points using the following bisection method. The upper and lower bounds are first determined from the spacing of the nodes, then the interval of possible γ is bisected by interpolating each value data point using other data points, and comparing the approximation error $\tilde{f}(x_{vi}) - f(x_{vi})$ with $\sigma(x_{vi})$ calculated using (15).

In determining the upper and lower bounds, we rely on the fact that the reciprocal of γ models the smallest length scale of f. The possible length scales that can be reconstructed from the finite number of data points are limited by the span of the data points on one end, and by the Nyquist frequency on the other end. Specifically, we start the bisection with

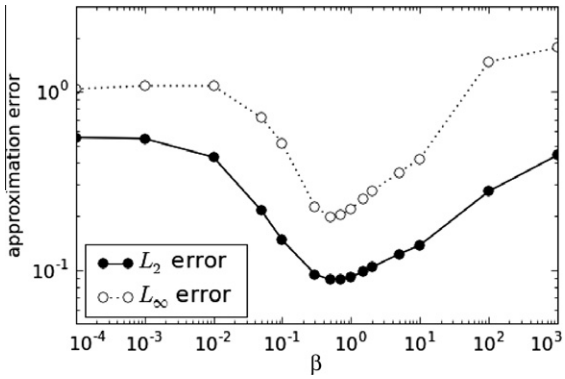

Fig. 6. Norms of the approximation error $\tilde{f}(x) - f(x)$ plotted against the magnitude β. The target function is $f(x) = x_1^2 + x_2^2$. The wavenumber parameter $\gamma = 2.0$.

$$\gamma_{\min} = \frac{1}{\delta_{\max}}, \quad \gamma_{\max} = \frac{\pi}{\delta_{\min}},$$

where $\delta_{\max}$ is the maximum distance between any two nodes, including value nodes and gradient nodes. $\delta_{\min}$ is the minimum distance between any two nodes.

The interval $[\gamma_{\min}, \gamma_{\max}]$ is then bisected in a logarithmic scale. At each step, set $\gamma_{\mathrm{mid}} = \sqrt{\gamma_{\min}\gamma_{\max}}$. For each $i = 1, \ldots, n_v$, we use our interpolation scheme to calculate $\tilde{f}(x_{vi})$ with $\gamma = \gamma_{\mathrm{mid}}$, using all data points other than the one at x_{vi}. We then compare $\sigma(x_{vi})$ with the approximation error $\tilde{f}(x_{vi}) - f(x_{vi})$. We decide that $\gamma < \gamma_{\mathrm{mid}}$ if $\sigma(x_{vi})$ are too large compared to the approximation errors, or $\gamma > \gamma_{\mathrm{mid}}$ if they are too small. Specifically, we set

$$\gamma_{\max} = \gamma_{\mathrm{mid}} \quad \text{if } \frac{1}{n}\sum_{i=1}^{n} \frac{(\tilde{f}(x_{vi}) - f(x_{vi}))^2}{\sigma(x_{vi})^2} < \frac{1}{C_F},$$

$$\gamma_{\min} = \gamma_{\mathrm{mid}} \quad \text{if } \frac{1}{n}\sum_{i=1}^{n} \frac{(\tilde{f}(x_{vi}) - f(x_{vi}))^2}{\sigma(x_{vi})^2} > \frac{1}{C_F},$$

where C_F is the conservative factor, which is usually set to 1. A larger C_F produces a larger γ, making the approximation scheme more conservative and less prone to oscillations, but less accurate for smooth functions on good quality grids.

When the measurement errors are nonzero $f(x_{vi})$ are unknown. In this case, we use a similar bisection criteria

$$\begin{aligned} &\gamma_{\max} = \gamma_{\mathrm{mid}} \quad \text{if } \frac{1}{n}\sum_{i=1}^{n} \frac{(\tilde{f}(x_{vi}) - f(x_{vi}))^2}{\sigma(x_{vi})^2 + \sigma_{vi}^2} < \frac{1}{C_F}, \\ &\gamma_{\min} = \gamma_{\mathrm{mid}} \quad \text{if } \frac{1}{n}\sum_{i=1}^{n} \frac{(\tilde{f}(x_{vi}) - f(x_{vi}))^2}{\sigma(x_{vi})^2 + \sigma_{vi}^2} > \frac{1}{C_F}. \end{aligned} \tag{17}$$

The bisection continues until $\gamma_{\min}$ and $\gamma_{\max}$ are sufficiently close.

We stop the bisection when

$$\frac{\gamma_{\max}}{\gamma_{\min}} < T_\gamma$$

for some threshold T_γ. At this point, we use γ_{mid} as the estimation for the "roughness" parameter γ. Through numerical experiments with a number of different functions, we found that $T_\gamma \approx 1.1$ is enough to produce very good results.

6. Numerical experiments

This section uses several numerical experiments to demonstrate the robustness and accuracy of our approximation scheme with automatically calculated parameters.

6.1. Using gradient data points

In the previous examples, we did not use any gradient data points, i.e., $n_g = 0$. In this section, we demonstrate that using gradient data points can greatly enhance the accuracy of the approximation function $\tilde{f}(x)$.

The results presented in this section are calculated in the "automatic" mode. Instead of being set by the user, the parameters β and γ are determined from the data points by Eqs. (16) and (17). These results also show that the methods of determining the parameters from the data points described in Section 5 work well.

Figs. 7 and 8 show the approximation results on a one-dimensional notch function

$$f(x) = \cos x - 2e^{-(4x)^2},$$

which is plotted as dashed lines in Figs. 7 and 8. In the cases of $n_v = n_g$, the gradient nodes are the same as the value nodes. As can be seen, the approximations constructed with $n_g = n_v = 16$ are as accurate as the approximation with $n_v = 24$, $n_g = 0$; while the approximation constructed with $n_g = n_v = 24$ are much more accurate. Fig. 9 shows the approximation results on the two-dimensional cosine function

$$f(x) = \cos x_1 + x_2,$$

whose contour consists of parallel diagonal straight lines. Fig. 10 shows the approximation results on the two-dimensional Runge function

$$f(x) = \frac{1}{1 + x_1^2 + x_2^2},$$

whose contour consists of concentric circles. As can be seen, for the same n_v, the contour lines of $\tilde{f}(x)$ with $n_g = n_v$ always look more similar to the contour lines of the target function than the contour lines of $\tilde{f}(x)$ constructed with $n_g = 0$. The conver-

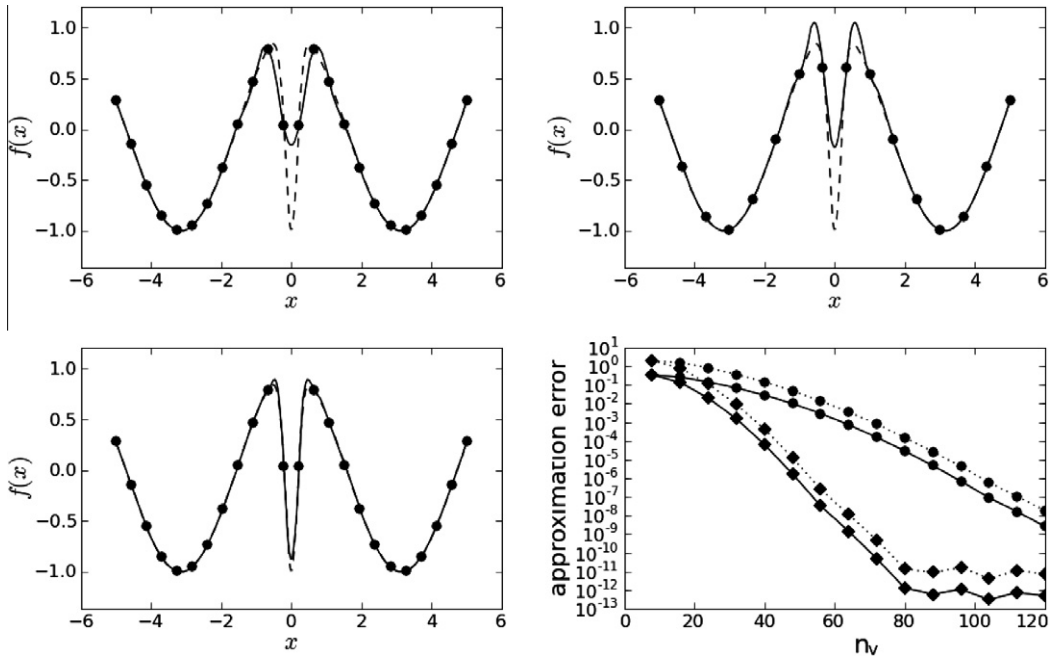

Fig. 7. Approximating the notched cosine function with uniform grids. The number of data points for each plot is $n_v = 24$, $n_g = 0$ (upper-left), $n_v = n_g = 16$ (upper-right), and $n_v = n_g = 24$ (lower-left). In the cases of $n_v = n_g$, the gradient nodes are the same as the value nodes. The target function is indicated by the dashed lines, and the solid lines represent the approximation function $\bar{f}$. The lower-right plot shows the convergence of the approximation error with respect to n_v. The circles indicates the case of $n_g = 0$; the diamonds indicates the case of $n_g = n_v$. The solid lines indicates L_2 error; the dotted lines indicates L_∞ error.

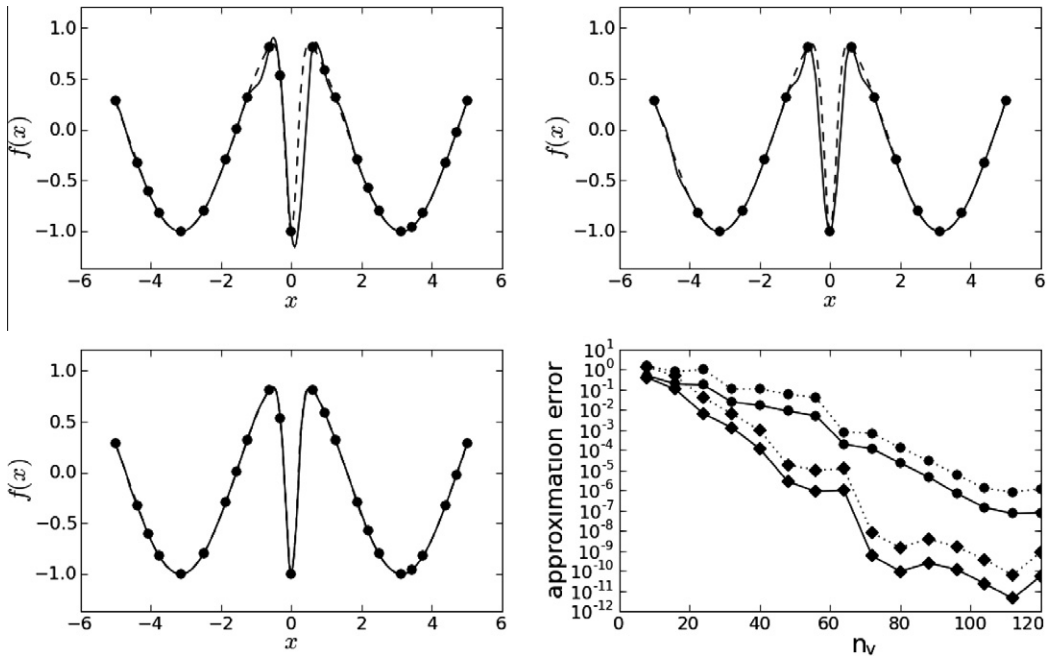

Fig. 8. Approximating the notched cosine function with quasi-random grids. The number of data points for each plot is $n_v = 24$, $n_g = 0$ (upper-left), $n_v = n_g = 16$ (upper-right), and $n_v = n_g = 24$ (lower-left). In the cases of $n_v = n_g$, the gradient nodes are the same as the value nodes. The lower-right plot shows the convergence of the approximation error.

gence plots in all four figures convey the same information, i.e., for the same n_v, the errors with $n_g = n_v$ are always less than the errors with $n_g = 0$.

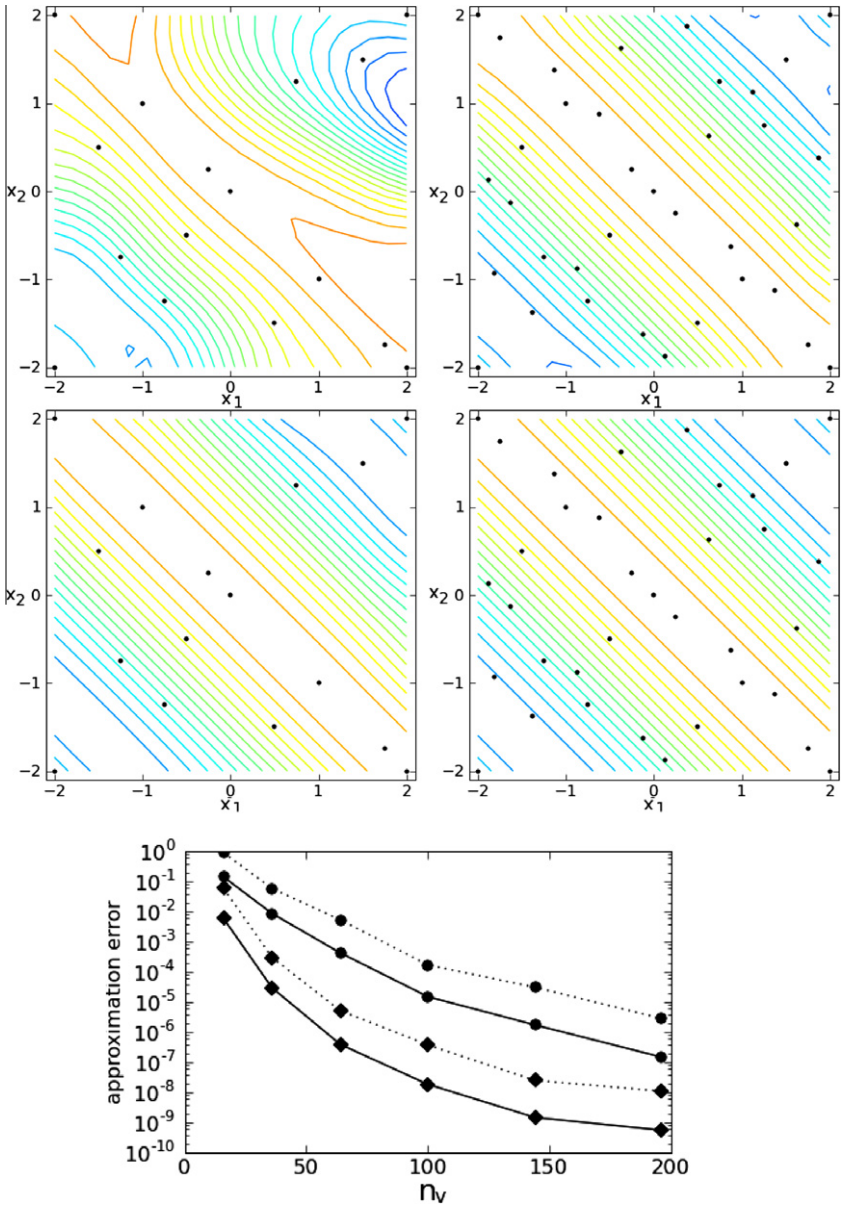

Fig. 9. Approximating the 2-D cosine function using quasi-random grids. The number of data points for each plot is $n_v = 16$, $n_g = 0$ (upper-left), $n_v = 36$, $n_g = 0$ (upper-right), $n_v = n_g = 16$ (middle-left), $n_v = n_g = 36$ (middle-right). In the cases of $n_v = n_g$, the gradient nodes are the same as the value nodes. The lower plot shows the convergence of the approximation error with respect to n_v (horizontal axis). The circles indicates the case of $n_g = 0$; the diamonds indicates the case of $n_g = n_v$. The solid lines indicates L_2 error; the dotted lines indicates L_∞ error.

6.2. Robustness to grid arrangement

Fig. 11 shows the behavior of the scheme when all the data points are located along a straight line. The contour lines indicate that the scheme is robust to the arrangement of data points. Fig. 12 demonstrates the extrapolatory behavior of the scheme. while all the data points lie on a parabola. The scheme extracts the negative y-component of the gradient from the data points, and creates a peak below the x-axis. Fig. 13 shows the contour lines of the interpolation function for a discontinuous function

$$f(x,y) = \begin{cases} +1, & x+y \geqslant 0, \\ -1, & x+y < 0. \end{cases}$$

All data points lies on the line $x = y$. The interpolation surface show Gibbs oscillations near the discontinuity. The magnitude of overshoot is comparable to that of Chebyshev polynomial approximation. Away from the data points, the jump is significantly smeared. In the three examples above, we use automatically calculated parameters as described in Section 5.

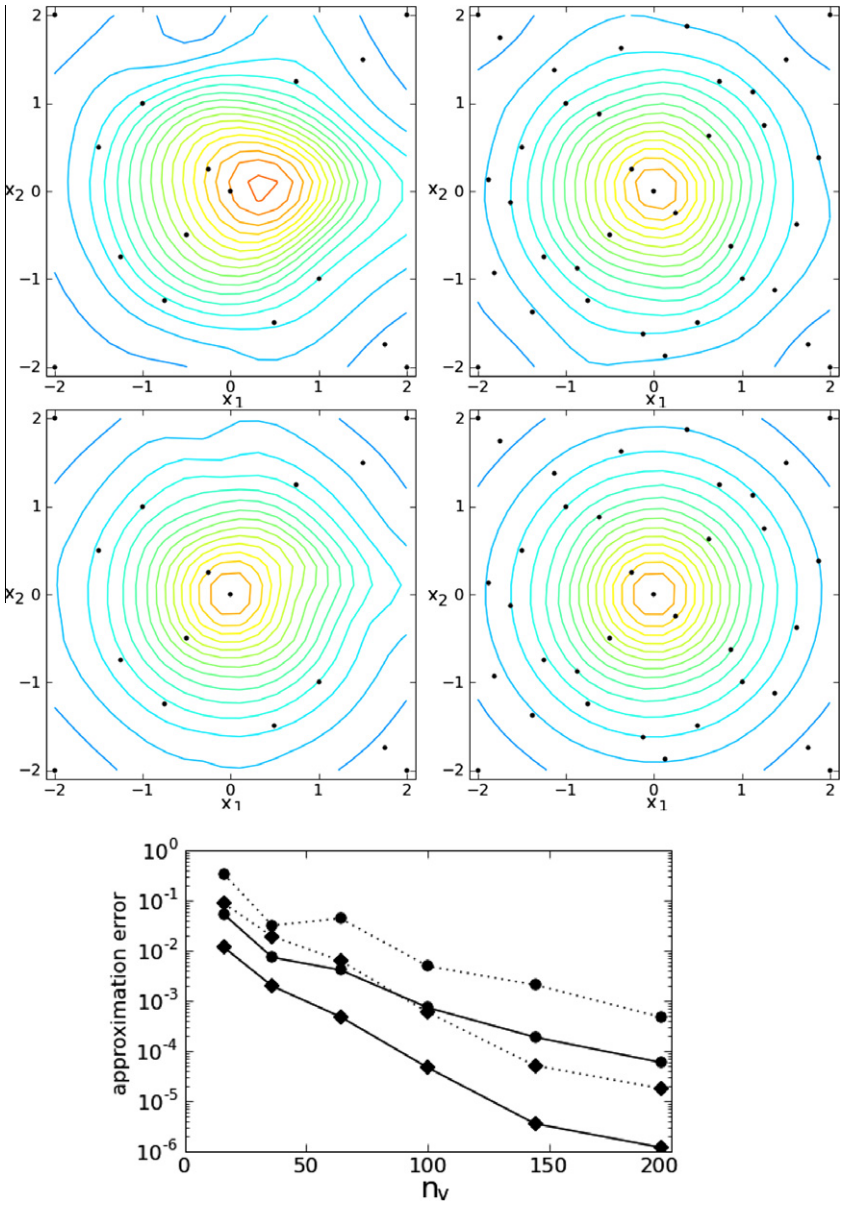

Fig. 10. Approximating the 2-D Runge function using quasi-random grids. The number of data points for each plot is $n_v = 16$, $n_g = 0$ (upper-left), $n_v = 36$, $n_g = 0$ (upper-right), $n_v = n_g = 16$ (middle-left), $n_v = n_g = 36$ (middle-right). In the cases of $n_v = n_g$, the gradient nodes are the same as the value nodes. The lower plot shows the convergence of the approximation error with respect to n_v (horizontal axis). The circles indicates the case of $n_g = 0$; the diamonds indicates the case of $n_g = n_v$. The solid lines indicates L_2 error; the dotted lines indicates L_∞ error.

Fig. 14 shows our interpolation surface for the Runge's function $1/(x_1^2 + x_2^2)$ with different grid arrangements. The L_2 error on the three grids are 0.004, 0.006 and 0.011, respectively; the L_∞ error are 0.039, 0.037 and 0.081, respectively. The parameters γ and β are automatically calculated.

6.3. A five-dimensional example

Fig. 15 shows the approximation error for the Runge's function in five-dimensional space:

$$f(x) = \frac{1}{1 + \sum_{k=1}^{5} x_k^2}, \quad x \in [-1, 1]^5.$$

The data points are the 5D Niederreiter sequence, and the accuracy is measured on the 1501st to 1600th points in the sequence. This example shows that our approximation scheme is scalable to high dimensions, and retains its high convergence rate.

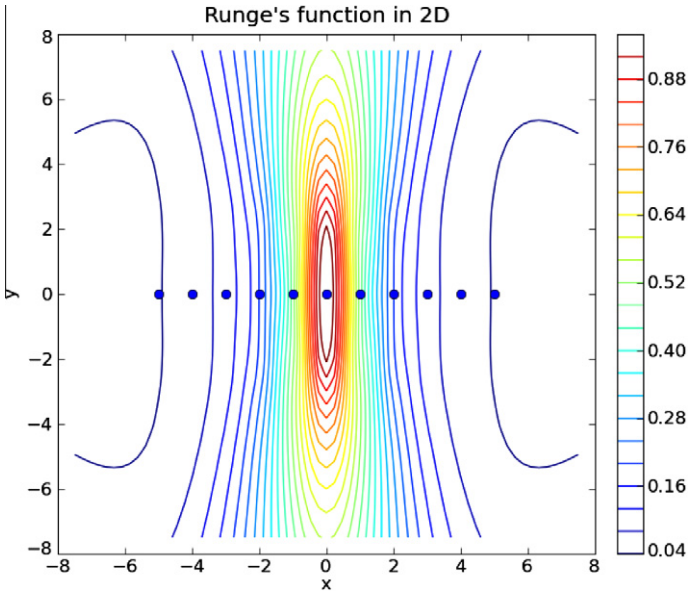

Fig. 11. Interpolation surface when all the data points are located along a straight line.

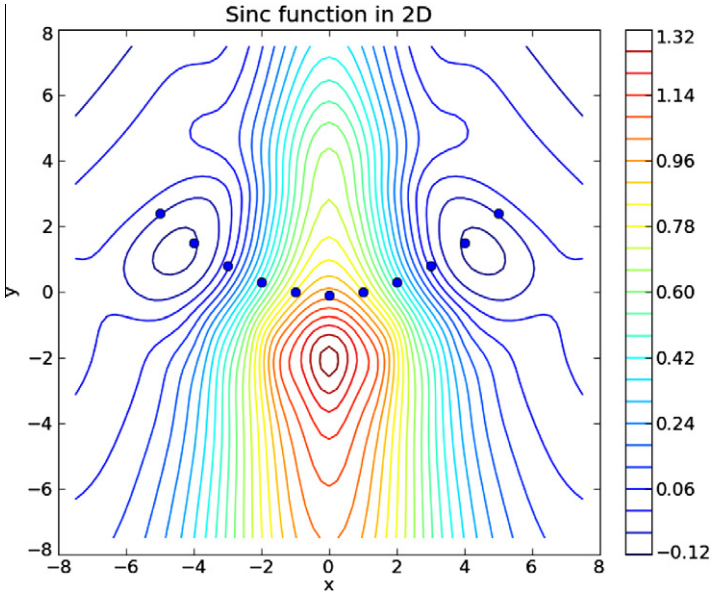

Fig. 12. Interpolation surface when all the data points are located along a parabola.

6.4. Comparison with kriging and Gaussian radial basis function

This section compares the accuracy and computational cost of our interpolation approximation against those of ordinary kriging [21,14] and Gaussian radial basis function [15,16,22] in both 2D and 3D. In the 2D comparison, the data points are the first n points in the 2D Niederreiter quasi-random sequence in the square domain $[-2,2]\times[-2,2]$. The accuracy of the schemes are accessed by evaluating the interpolant on the 501st to 600th points of the quasi-random sequence. The L_∞ error of the approximation is estimated by the largest approximation error on the 100 points; the L_2 error is estimated by the root mean square of the approximation errors. In the 3D comparison, the data points are the 3D Niederreiter sequence, and the accuracy is measured on the 1501st to 1600th points in the sequence.

From Fig. 16, it is clear that our approximation has higher rate of convergence than ordinary kriging with exponential correlation function. When the number of data points is large enough, the approximation error of our scheme is much smaller. Fig. 17 compares our approximation scheme with Gaussian radial basis function with kernels $\exp(-\gamma r^2/2)$, where

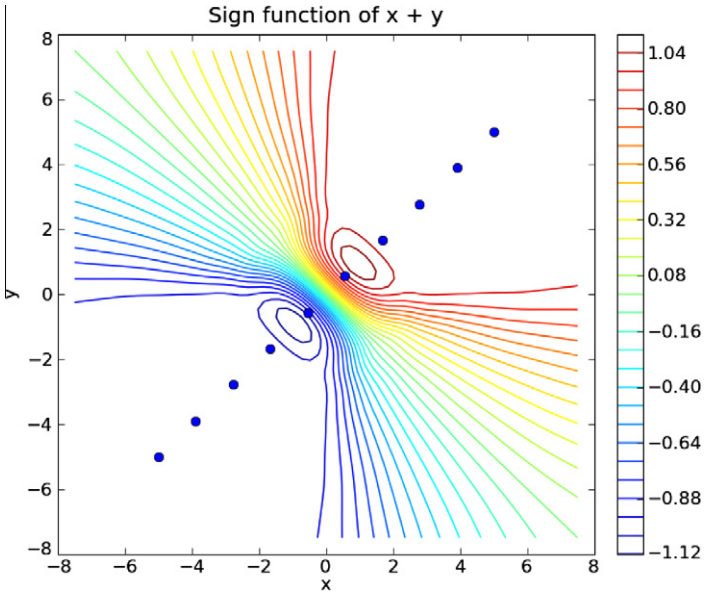

Fig. 13. Interpolation surface of a discontinuous function.

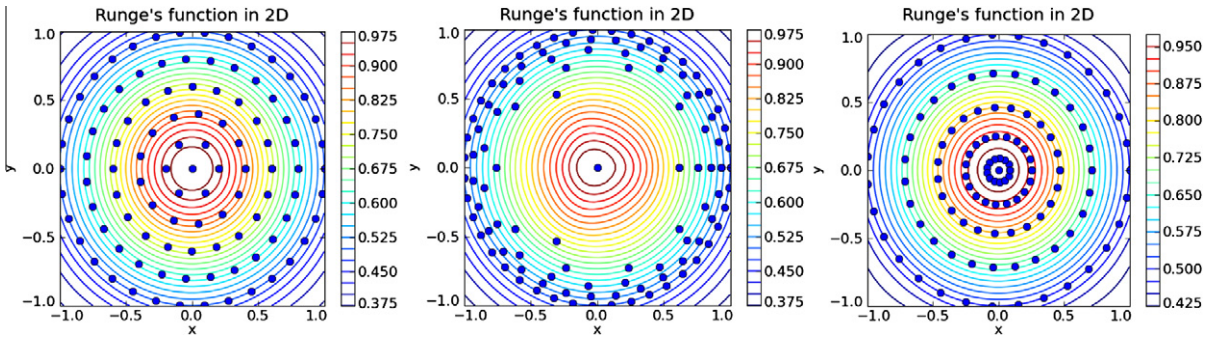

Fig. 14. Interpolating the Runge's function with different grid arrangements. The number of data points are 106, 112 and 119, respectively.

different colors represent the L_2 error of different γ. For smaller γs, the radial basis function approximation is more accurate for small number of data points. As the number of data points increase, the relative accuracy of radial basis function with small γ deteriorates. This may be a result of the Runge's phenomenon, as the refined data points resolve higher frequency oscillations of the target function. Radial basis function approximation with larger γ has larger approximation errors for small number of data points; but its error continue to decrease exponentially as data points refine. Our approximation scheme with automatically calculated parameters has consistent near-optimal performance across the range of data points. In the 2D comparison, our scheme is about 5 times more accurate than Gaussian radial basis function with the optimal shape function for large number of data points.

Fig. 18 compares the computational cost of our approximation scheme with ordinary kriging and Gaussian radial basis function. The computations are performed on a 2.13 GHz Intel Xeon processor, using the GNU Scientific Library for linear algebra calculations. The computation time of our method is what it takes to calculate the values of all basis functions a_{vi} at each points, while the computation time of kriging and Gaussian RBF is what it takes to construct the interpolant. The computation cost for evaluating the kriging and RBF interpolants are negligible. This comparison shows that our method takes significantly more computation time. When the approximation function needs to be evaluated at many points, the computation cost of our method is proportional to the number of points, while evaluating additional points for kriging and RBF is essentially free. The operation count of our approximation scheme is discussed in detail in Section 7.

7. Discussion of computation cost, limitations and potential applications

Our approximation scheme solves a constrained least squares problem (12) for each point where the approximation function is evaluated. The computation cost for solving this least squares problem is $O(N_C N_R^2)$, where

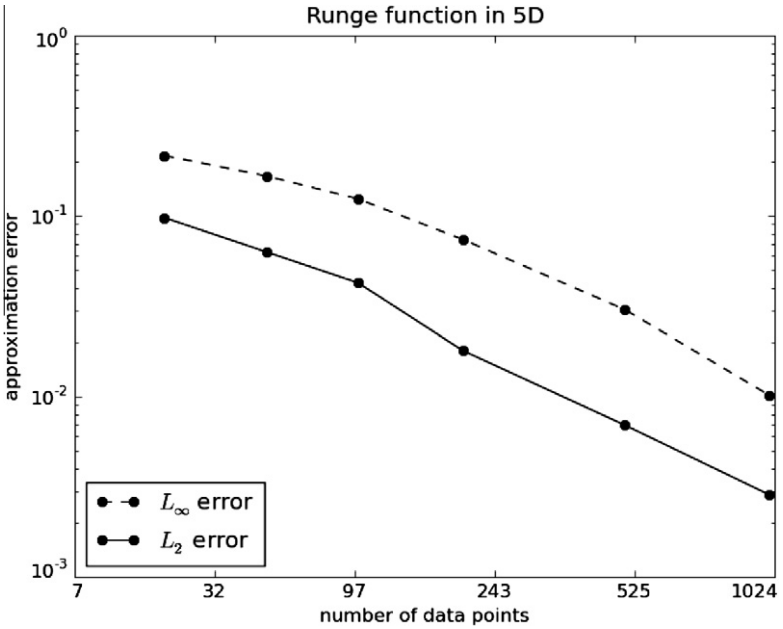

Fig. 15. Interpolation error for the Runge's function in 5D.

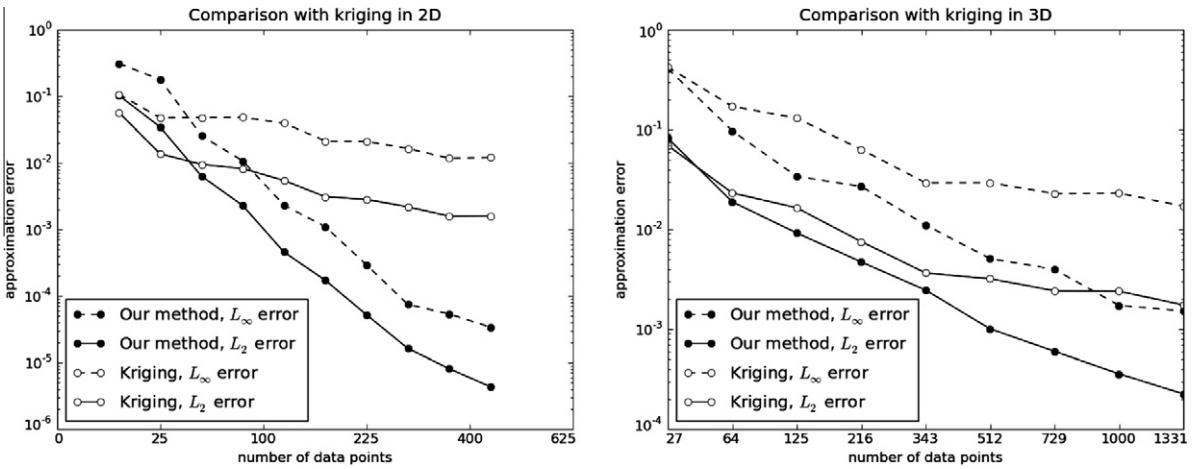

Fig. 16. Comparison of interpolation accuracy with ordinary kriging with exponential correlation function.

$$N_R = n_v + dn_g$$

is the number of columns of the matrix involved, which is equal to the degree of freedom provided by the data points;

$$N_C = |\{\kappa : |\kappa| < N\}| + N_R$$

is the number of rows of the matrix. Since N is chosen according to Eq. (4), we have $N_C \approx 2N_R$. Therefore, the computation cost for evaluating each point is $O((n_v + dn_g)^3)$.

Moreover, the algorithm for calculating the parameter γ described in Section 5 adds another $O(Kn_v(n_v + dn_g)^3)$ operations, where K is the number of bisection iterations. This makes the total computation cost for evaluating the approximation function at L points the order of $O((K\, n_v + L)\,(n_v + dn_g)^3)$.

Besides the rapidly increasing computation cost, the condition number of the least squares system (12) is another concern. The matrix involved in the least squares system (12) resembles a Vandermonde matrix, which is known to be ill-conditioned at large size. Therefore, numerical issue will arise when the number of data points exceeds around 1000. In addition, when two data points are very close to each other, the corresponding columns of the matrix will become almost linearly

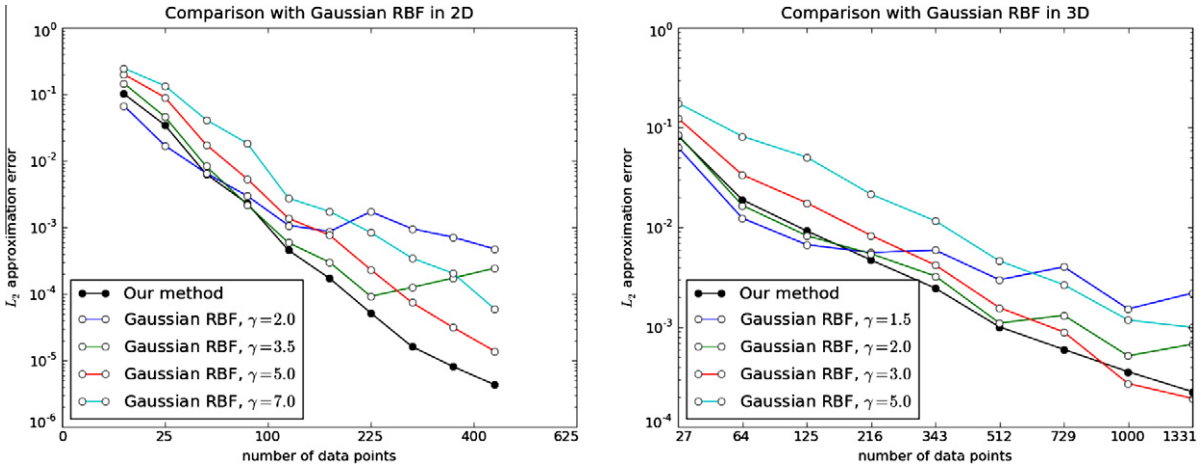

Fig. 17. Comparison of interpolation accuracy with Gaussian radial basis function interpolation.

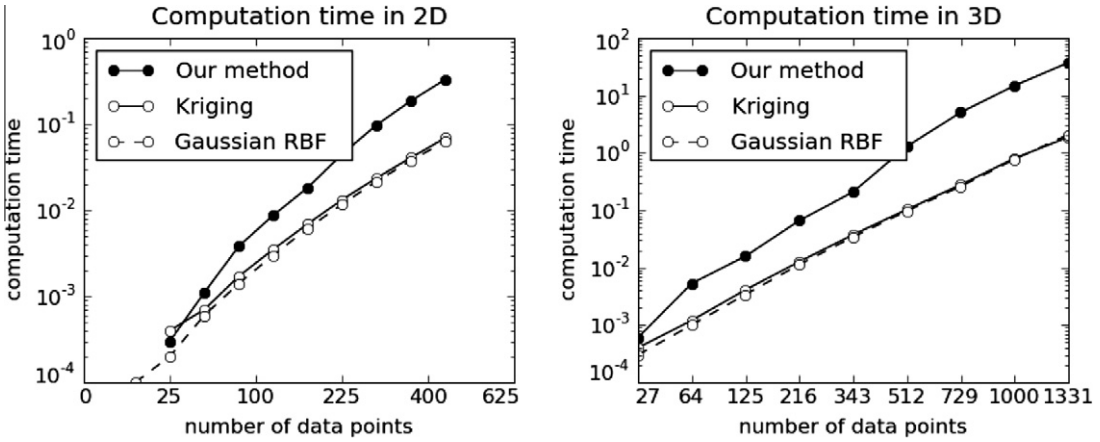

Fig. 18. Comparison of computation time with ordinary kriging and Gaussian RBF.

dependent, making the matrix almost singular. Although our scheme works well for grids shown in Fig. 14, more extreme point distributions will introduce large round-off errors in the approximation function.

The operation counts show that our scheme is more computationally expensive than most existing schemes; the condition number of the least squares system limits the number of data points. Nevertheless, our scheme is useful in some applications. A motivational example for the development of our approach is response surface modeling [23]. In this case, each data point is obtained by solving a computationally intensive mathematical model such as a partial differential equation. The total number of data points often does not exceed 1000. Each data point may contain different amount of uncertainty, which can be estimated from the final residual of the evaluation. Gradient information may be available in the form of adjoint solutions. Because obtaining the function value and gradient at each data point is computationally intensive, using our scheme to construct a response surface often adds little to the total computation time. Compared to computation cost, accuracy is a more important consideration in this application. Our approximation method is suitable for such applications.

8. Software implementation

This multivariate approximation scheme was implemented in an open source software package available under GNU General Public License. The implementation is in C, using GNU Scientific Library for matrix manipulation and linear algebra. Interfaces for three different languages, C, Fortran and Python, are supported. The source code can be downloaded at http://web.mit.edu/qiqi/www/mir/.

9. Conclusion

We have developed a multivariate interpolation and regression scheme. The scheme produces an interpolation surface when the data points are given as exact, or a nonlinear regression function when nonzero measurement errors are associated

with the data points. The resulting approximation features high order convergence, and is robust for a variety of node distributions. Both parameters of the scheme, β and γ, can be automatically determined based on the data points. An uncertainty bound of the approximation function is also produced by the scheme. When the gradient measurement are available at some nodes, they can be used to greatly improve the accuracy of the approximation.

Acknowledgment

This work was funded by the United States Department of Energy's PSAAP Program at Stanford University.

References

[1] E. Meijering, A chronology of interpolation: from ancient astronomy to modernsignal and image processing, Proceedings of the IEEE 90 (3) (2002) 319–342.
[2] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods: Fundamentals in Single Domains, Springer, 2006.
[3] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics, Springer, 2007.
[4] L. Dufresne, G. Dumas, A spectral/b-spline method for the Navier–Stokes equations in unbounded domains, J. Comput. Phys 185-2 (2003) 532–548.
[5] P.P. Chinchapatnam, K. Djidjeli, P.B. Nair, Radial basis function meshless method for the steady incompressible Navier–Stokes equations, Int. J. Comput. Math. 84 (10) (2007) 1509–1521.
[6] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic pdes, Comput. Math. Appl 46 (2003) 891–902.
[7] L. Mathelin, M. Hussaini, T. Zang, Stochastic approaches to uncertainty quantification in cfd simulations, Numer. Algorithms 38-1 (2005) 209–236.
[8] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139. <http://dx.doi.org/10.1137/040615201>.
[9] C.W. Borchardt, Interpolationsformene interpolationsformel für eine art symmetrischer functionen und über deren anwendung, in: Abhandlungen der Königlichen Akademie der Wissenschaften zu, Berlin, 1860, pp. 1–20.
[10] L. Kronecker, Über einige interpolationsformeln für ganze functionen mehrer variabeln, in: Monatsberichte der Königlich Preussischen Akademie der Wissenschaften zu, Berlin, 1865, pp. 686–691.
[11] C. de Boor, A. Ron, The least solution for the polynomial interpolation problem, Math. Z. 210 (1) (1992) 347–378.
[12] A. Cuyt, B. Verdonk, Multivariate rational interpolation, Computing 34 (1) (1985) 41–61.
[13] W. Gordon, J. Wixom, Shepard's method of "metric interpolation" to bivariate and multivariate interpolation, Math. Comput. 32 (141) (1978) 253–264.
[14] H. Wackernagel, Multivariate Geostatistics – An Introduction with Applications, Springer, Berlin, 1995.
[15] M.J.D. Powell, Radial basis functions for multivariable interpolation: a review, in: Algorithms for Approximation, Clarendon Press, New York, NY, USA, 1987, pp. 143–167.
[16] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in rbf interpolation, Comput. Math. Appl. 54 (3) (2007) 379–398.
[17] R. Franke, Scattered data interpolation: tests of some methods, Math. Comput. 38 (157) (1982) 181–200.
[18] G.E. Fasshauer, Meshfree Approximation Methods using MATLAB, World Scientific Publishers, 2007.
[19] Q. Wang, P. Moin, G. Iaccarino, A rational interpolation scheme with super-polynomial rate of convergence, SIAM J. Numer. Anal. 47 (6) (2009) 4073–4097.
[20] G.H. Golub, C.F.V. Loon, Matrix Computations, The Johns Hopkins University Press, Baltimore, 1996.
[21] P. Poovaerts, Geostatistics for Natural Resources Evaluation, Oxford University Press, 1997.
[22] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. (2004) 853–867.
[23] Constructing Response Surfaces Using Imperfect Function Evaluations, AIAA-2010-2925, 2010.