**Journal of Advances in Mathematics and Computer Science**

**29(3): 1-10, 2018; Article no.JAMCS.44695**

*ISSN: 2456-9968*

*(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)*

# A Multilevel LSSVM for Elliptic Partial Differential Equations

**Lin Dong[1*] and Jie Liao[1]**

[1] *East China University of Science and Technology, Shanghai 200237, P.R. China.*

***Authors' contributions***

*This work was carried out in collaboration between both authors. Author JL designed the study, carried out the model analysis and wrote the first draft of the manuscript. Author LD managed the numerical simulation of the study and assisted to fine tune the model. Both authors read and approved the final manuscript.*

***Article Information***

DOI: 10.9734/JAMCS/2018/44695

Editor(s):

(1)Dr. Jacek Dziok, Professor, Institute of Mathematics, University of Rzeszow, Poland.

Reviewers:

(1)Zhiliang Deng, University of Electronic Science and Technology of China, China.

(2)Aliyu Bhar Kisabo, Nigeria.

Complete Peer review History: http://www.sciencedomain.org/review-history/26877

***Received: 12 August 2018***

***Accepted: 20 October 2018***

***Published: 27 October 2018***

***Original Research Article***

## Abstract

In this paper, we propose a multilevel least square support vector machine (LSSVM) for solving elliptic boundary value problems based on wavelet kernel functions. This algorithm is constructed by a sequence of residual corrections and separating the computations of different levels, where different scale parameters are employed to accommodate different scales. In this multilevel algorithm, a coarse data set and a large scale parameter are chosen and the target function is interpolated in this data set to capture the large-scale variations of the target function at the first level, next, a smaller scale parameter is used to interpolate the residuals on a finer data set, capturing the finer details on the second level. The numerical tests on some linear second order elliptic boundary value problems show the efficiency of the multilevel algorithm.

*Keywords: Wavelet kernel function; least square support vector machine; multilevel.*

**2010 Mathematics Subject Classification:** 35H10, 76P99, 82B21

**Corresponding author: E-mail: ldong_ecust@163.com;*

# 1 Introduction

There are many machine learning methods [1] for solving differential equations originating from neural network approaches that have been wildly studied in recent years [2, 3, 4, 5]. In this paper, we discusse the Least Square Support Vector Machine (LSSVM, [6, 7]) based learning method for solving differential equations [8]. Different from traditional mesh-based numerical methods, the LSSVM does not need any mesh topology and it can give an approximate solution in closed form (continuous and differentiable), desired accuracy for PDEs with irregular domains also can be achieved.

Learning solutions to partial differential equations using radial base function(RBF) kernel based LSSVM have proven to be particularly powerful. One important issue of this approach is with respect to the scale parameter (support radius) in the kernel function. Numerical experience shows that, for small scales, the training of the network is generally a good conditioned optimization problem, but the general accuracy is low [9, 10], on the other side, for large scales, the training may become ill-conditioned, but the general accuracy is good with smooth interpolation behaviour. For these results, it is necessary to construct the multiscale RBF (MSRBF) network to accommodate both the local and the global properties of the basis functions by including both small and large scales in the network in a hierarchical multiscale way [9, 11, 12].

For the choice of kernel, a variety of RBF kernel could be possible, such as Gaussian radial base function. Note that the idea of multiscale RBF network somehow comes from wavelet decompositions [13, 14] and the nature of wavelet decompositions is multiscale. The wavelet methods derived from wavelet decompositions have be used for the numerical solution of differential equations, see [5, 15, 16]. And wavelet kernel support vector machine is shown to be successful in applications such as regression and classification, see [17, 18, 19, 20, 21, 22]. So compare with Gaussian RBF based LSSVM for solving differential equations proposed in [8], wavelet kernel functions are more suitable for our multilevel LSSVM to approximate a solution with localized feathers or structures. In this paper, we propose a wavelet kernel function based LSSVM for solving differential equations.

# 2 Multilevel Algorithm of LSSVM

## 2.1 LSSVM regression and wavelet kernel function

Consider a given training set $\{z_i, y_i\}_{i=1}^N$ with input data $z_i \in \mathbb{R}^m$ and output data $y_i \in \mathbb{R}$. Firstly, we recall the goal in regression is to estimate an approximation $\widehat{y} = w^{\mathrm{T}}\varphi(z) + d$, where $w \in \mathbb{R}^n$, $d \in \mathbb{R}$, $\varphi(\cdot) : \mathbb{R}^m \to \mathbb{R}^n$ is the nonlinear feature map and $n$ is the dimension of the feature space. The primal LSSVM model for regression is written as follows [7]:

$$\begin{array}{ll} \underset{w.d.e}{\text{minimize}} & \dfrac{1}{2}w^T w + \dfrac{\gamma}{2}e^T e \\ \text{subject to} & y_i = w^T\varphi(z_i) + d + e_i, \quad i = 1, ..., N, \end{array} \tag{2.1}$$

where $w$ and $d$ are parameters to be determined, $e = y - \widehat{y}$ is the error of the approximation and $\gamma \in \mathbb{R}^+$ is a regularization parameter. The dual solution is given by

$$\begin{bmatrix} \mathcal{K} + \gamma^{-1} I_N & 1_N \\ 1_N^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ d \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{2.2}$$

where $\mathcal{K}_{ij} = K(z_i, z_j) = \varphi(z_i)^T\varphi(z_j)$ is the $(i, j)$-th entry of the positive definite kernel matrix for a positive definite kernel function $K(z, r) = \varphi(z)^T\varphi(r)$ and $\alpha = [\alpha_1; ...; \alpha_N]$ is the vector of Lagrange multipliers. $1_N = [1; ...; 1] \in \mathbb{R}^N$, $y = [y_1; ...; y_N]$ and $I_N$ is the identity matrix.

The key issue of the support vector machine is the choosing of the kernel function. In fact, a function meets the Mercer conditions can be an admissible support vector kernel function [23]. In this paper, we choose Mexican hat mother wavelet

$$h(z) = (1 - z^2)\exp(-z^2/2). \tag{2.3}$$

Although not related, we note that a wavelet bases can be generated by taking dilations and translations of the mother wavelet function. Note that a scale parameter is involved in the dilation, thus can be naturally used to accommodate the multiscale phenomenon. Then an admissible wavelet support vector kernel function can be constructed.

**Theorem 2.1** ([17, 21]). *The Mexican hat wavelet kernel function, with scale parameter a,*

$$K_a(z_1, z_2) = K_a(z_1 - z_2) = \prod_{i=1}^{m} h\left(\frac{z_{1i} - z_{2i}}{a}\right) = \prod_{i=1}^{m}\left(1 - \frac{\|z_{1i} - z_{2i}\|^2}{a^2}\right)\exp\left(-\frac{\|z_{1i} - z_{2i}\|^2}{2a^2}\right), \tag{2.4}$$

*is a translation invariant admissible support vector kernel function. Here* $z_1, z_2 \in \mathbb{R}^m$, $a > 0$ *is called the scale parameter of the kernel function.*

## 2.2 Multilevel algorithm

We derive a multilevel algorithm by constructing a sequence of residual corrections and separating the computations of different levels. Constructe the optimization problem on each scale separately, so that the multilevel algorithm is proceeded by solving a sequence of optimization problem.

Given a linear second order elliptic partial differential equation

$$Lu(z) = f(z), \quad z \in \Sigma \subset \mathbb{R}^m, \tag{2.5}$$

with the boundary conditions on the boundary $\partial\Sigma$

$$Bu(z) = g(z), \quad z \in \partial\Sigma, \tag{2.6}$$

where $L$ and $B$ are linear second order elliptic partial differential operator and boundary operator respectively.

Let $K_k(z_1, z_2) = \varphi_k(z_1)^T\varphi_k(z_2)$ be kernel functions with different scale parameters on scale $k$, and $k = I, II.III, ...$ represent different scales. Suppose that a general approximate solution to (2.5)-(2.6) is of the form

$$\hat{u}(z) = \hat{u}_I(z) + \hat{u}_{II}(z) + \hat{u}_{III}(z) + ... = w_I^T\varphi_I(z) + d + w_{II}^T\varphi_{II}(z) + w_{III}^T\varphi_{III}(z) + ....$$

where

$$\hat{u}_I(z) = w_I^T\varphi_I(z) + d, \tag{2.7}$$

$$\hat{u}_k(z) = w_k^T\varphi_k(z), k = II, III, ... \tag{2.8}$$

To obtain the training set, assume a discretization of the domain $\Sigma$ into two sets of collocation points defined by

$$\mathcal{Z}_{\mathcal{D}}^k = \{z_{\mathcal{D}}^i \mid i = 1, \cdots, N_k\}, k = I, II, III, ...$$
$$\mathcal{Z}_{\mathcal{B}}^k = \{z_{\mathcal{B}}^i \mid i = 1, \cdots, M_k\}, k = I, II, III, ...$$

where $\mathcal{Z}_{\mathcal{D}}^k$ and $\mathcal{Z}_{\mathcal{B}}^k$ denote the sets of collocation points located inside the domain and the collocation points situated on the boundary, respectively. $N_k = |\mathcal{Z}_{\mathcal{D}}^k|$ and $M_k = |\mathcal{Z}_{\mathcal{B}}^k|$ denote the number of points in $\mathcal{Z}_{\mathcal{D}}$ and $\mathcal{Z}_{\mathcal{B}}$ on different scales are given.

For the first scale, the approximate solution can be found by solving the optimization problem

$$\begin{aligned} \underset{w_I,d,e_I}{\text{minimize}} \quad & \frac{1}{2}w_I^T w_I + \frac{\gamma}{2}e_I^T e_I \\ \text{subject to} \quad & L\hat{u}_I(z_{\mathcal{D}}^i) = f(z_{\mathcal{D}}^i) + e_{I,i}, \quad i = 1, ..., |\mathcal{Z}_{\mathcal{D}}^I|, \\ & B\hat{u}_I(z_{\mathcal{B}}^j) = g(z_{\mathcal{B}}^j), \quad j = 1, ..., |\mathcal{Z}_{\mathcal{B}}^I|. \end{aligned} \tag{2.9}$$

Where $e_I$ is the error on first scale. Let $\{\alpha_i^I\}_{i=1}^{|\mathcal{Z}_{\mathcal{D}}^I|}$ and $\{\beta_j^I\}_{j=1}^{|\mathcal{Z}_{\mathcal{B}}^I|}$ be Lagrange multipliers of the optimization problem (2.9). According the Karush-Kuhn-Tucker(KKT) optimality conditions, we have the dual problem in the following Lemma.

**Lemma 2.2.** *[8] Suppose that $K_I$ is the Mexican hat wavelet kernel function defined in* (2.4) *with $K_I(z_1, z_2) = \varphi_I(z_1)^T\varphi_I(z_2)$ and $\gamma_I \in \mathbb{R}^+$ is a regularization parameter, then the dual problem to* (2.9) *for solving* (2.5)-(2.6) *is given by*

$$\begin{bmatrix} \mathcal{K}_{\mathcal{I}} + \gamma_I^{-1} I_N & S_I & c \cdot 1_{N_1} \\ S_I^T & \triangle_I & \tilde{c} \cdot 1_{M_1} \\ c \cdot 1_{N_1}^T & \tilde{c} \cdot 1_{M_1}^T & 0 \end{bmatrix} \begin{bmatrix} \alpha_I \\ \beta_I \\ d \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}. \tag{2.10}$$

*The elements in* (2.10) *are given by:*

$$\begin{aligned} & N_1 = |\mathcal{Z}_{\mathcal{D}}^I|, \quad M_1 = |\mathcal{Z}_{\mathcal{B}}^I|, \\ & \mathcal{K}_I = (K_{ij}^I)_{N\times N}, \quad K_{ij}^I = [L(\varphi_I(z_{\mathcal{D}}^i))]^T[L(\varphi_I(z_{\mathcal{D}}^j))], \\ & S_I = (s_{ij}^I)_{N_1\times M_1}, \quad S_{ij}^I = [B(\varphi_I(z_{\mathcal{B}}^i))]^T[L(\varphi_I(z_{\mathcal{D}}^j))], \\ & \triangle_I = (\triangle_{ij}^I)_{M_1\times M_1}, \quad \triangle_{ij}^I = [B(\varphi_I(z_{\mathcal{B}}^i))]^T[B(\varphi_I(z_{\mathcal{B}}^j))], \\ & c = L(d)/d, \quad \tilde{c} = B(d)/d, \\ & \alpha_I = [\alpha_1^I, \cdots, \alpha_{N_1}^I]^T, \quad \beta_I = [\beta_1^I, \cdots, \beta_{M_1}^I]^T, \\ & f = [f(z_{\mathcal{D}}^1), \cdots, f(z_{\mathcal{D}}^{N_1})]^T, \ g = [g(z_{\mathcal{B}}^1), \cdots, g(z_{\mathcal{B}}^{M_1})]^T. \end{aligned}$$

*The approximate solution to* (2.5)-(2.6) *is given by*

$$\hat{u}(z) = \hat{u}_I(z) = d + \sum_{i=1}^{|\mathcal{Z}_{\mathcal{D}}|} \alpha_i [L(\varphi(z_{\mathcal{D}}^i))]^T\varphi(z) + \sum_{i=1}^{|\mathcal{Z}_{\mathcal{B}}|} \beta_i [B(\varphi(z_{\mathcal{B}}^i))]^T\varphi(z). \tag{2.11}$$

Then the second scale is a correcting to the first scale. The third or higher level scales can be also added to improve the accuracy. For $k$-th($k = II, III, ...$) scale, define the residuals by

$$\widetilde{f} = f - L\hat{u}_I - ... - L\hat{u}_{k-1}, \qquad \widetilde{g} = g - B\hat{u}_I - ... - B\hat{u}_{k-1},$$

therefore the problem becomes

$$\begin{cases} L\widetilde{u}(z) = \widetilde{f}(z), & z \in \Sigma, \\ B\widetilde{u}(z) = \widetilde{g}(z), & z \in \partial\Sigma. \end{cases} \tag{2.12}$$

And define the error by

$$e_k = u - \hat{u}_I - ... - \hat{u}_k$$

The residuals need to be interpolated at a finer data set on scale $k$, where $\alpha_k$ and $\beta_k$ can be obtained by the optimization problem:

$$\begin{aligned} \underset{w_k.e_k}{\text{minimize}} \quad & \frac{1}{2}w_k^T w_k + \frac{\gamma_k}{2}e_k^T e_k \\ \text{subject to} \quad & L\hat{u}_k(z_{\mathcal{D}}^i) = \widetilde{f}(z_{\mathcal{D}}^i) + e_{k,i}, \quad z_{\mathcal{D}}^i \in \mathcal{Z}_{\mathcal{D}}^k, \ i = 1, \cdots, |\mathcal{Z}_{\mathcal{D}}^k|, \\ & B\hat{u}_k(z_{\mathcal{B}}^j) = \widetilde{g}(z_{\mathcal{B}}^j), \quad z_{\mathcal{B}}^j \in \mathcal{Z}_{\mathcal{B}}^k, \ j = 1, \cdots, |\mathcal{Z}_{\mathcal{B}}^k|, \end{aligned} \tag{2.13}$$

**Lemma 2.3.** *Suppose that $K_k$ are the Mexican hat wavelet kernel functions with different scale parameters defined in* (2.4) *with $K_k(z_1, z_2) = \varphi_k(z_1)^T\varphi_k(z_2)$ and $\gamma_k \in \mathbb{R}^+$ are regularization parameters, let $\alpha_i$, $\beta_i$ and $d$ be obtained on scale $i(i = I, II, ..., k-1)$. Then the parameters $\alpha_k$ and $\beta_k$ on scale $k$ satisfy the dual problem:*

$$\begin{bmatrix} \mathcal{K}_k + \gamma_k^{-1} I_{N_2} & S_k \\ S_k^T & \triangle_k \end{bmatrix} \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \widetilde{f} \\ \widetilde{g} \end{bmatrix}. \tag{2.14}$$

*The elements of* (2.14) *are given by:*

$$\begin{aligned}
&\mathcal{K}_k = (K_{ij}^k)_{N_k \times N_k}, \quad K_{ij}^k = [L(\varphi_k(z_{\mathcal{D}}^i))]^T [L(\varphi_k(z_{\mathcal{D}}^j))], \\
&S_k = (S_{ij}^k)_{N_k \times M_k}, \quad S_{ij}^k = [B(\varphi_k(z_{\mathcal{B}}^i))]^T [L(\varphi_k(z_{\mathcal{D}}^j))], \\
&\triangle_k = (\triangle_{ij}^k)_{M_k \times M_k}, \quad \triangle_{ij}^k = [B(\varphi_k(z_{\mathcal{B}}^i))]^T [B(\varphi_k(z_{\mathcal{B}}^j))], \\
&\alpha_k = [\alpha_1^k, \cdots, \alpha_{N_k}^k]^T, \quad \beta_k = [\beta_1^k, \cdots, \beta_{M_k}^k]^T, \\
&\widetilde{f} = [\widetilde{f}(z_{\mathcal{D}}^1), \cdots, \widetilde{f}(z_{\mathcal{D}}^{N_k})]^T, \ \widetilde{v} = [\widetilde{g}(z_{\mathcal{B}}^1), \cdots, \widetilde{g}(z_{\mathcal{B}}^{M_k})]^T,
\end{aligned}$$

*where $N_k = |\mathcal{Z}_{\mathcal{D}}^k|$, $M_k = |\mathcal{Z}_{\mathcal{B}}^k|$, and the approximate solution to* (2.5)-(2.6) *is given by*

$$\begin{aligned}
\hat{u}(z) &= \hat{u}_I(z) + ... + \hat{u}_{k-1}(z) + \hat{u}_k(z) \\
&= \hat{u}_I(z) + ... + \hat{u}_{k-1}(z) + \sum_{i=1}^{|\mathcal{Z}_{\mathcal{D}}^k|} \alpha_i^k [L(\varphi_k(z_{\mathcal{D}}^i))]^T \varphi_k(z) + \sum_{i=1}^{|\mathcal{Z}_{\mathcal{B}}^k|} \beta_i^k [B(\varphi_k(z_{\mathcal{B}}^i))]^T \varphi_k(z).
\end{aligned}$$

Note that in the above superposition of scale, firstly, the coarse data set and the large scale parameter are used to capture the large-scale variations of the target function. Then the smaller scale parameter is used to interpolate the residuals on the finer data set, capturing now the finer details. The sum of both interpolants obviously better interpolates the target function at the data sites on the finer data set. This process can be further applied to finer and finer scales till the anticipated accuracy is achieved.

*Remark* 2.1. We restrict our attention to linear problem (2.5)-(2.6), $c = L(d)/d$, the action of the linear differential operator $L$ on the constant $d$, then divided by $d$, will be a constant, this can be easily understood, for example, if we take

$$L = a_1 \frac{\partial^2}{\partial x^2} + a_2 \frac{\partial^2}{\partial x \partial y} + a_3 \frac{\partial^2}{\partial y^2} + b_1 \frac{\partial}{\partial x} + b_2 \frac{\partial}{\partial y} + c,$$

thus the dual problem (2.10) is a linear system. As the same to [8], the computation of $k_{ij}$, $s_{ij}$ and $\triangle_{ij}$ can be converted to partial derivatives of the kernel function (2.4). With $(\alpha, \beta, d)$ be solved from (2.10), an explicit representation of the solution is achieved in the form (2.11).

# 3 Numerical Simulations

In this section, we will do some numerical simulations on both regular and irregular boundaries to test the multilevel algorithm introduced. The error of an approximate solution is measured by norms

$$\text{RMSE} = \sqrt{\sum_i e_i^2 / N}, \quad L_\infty = \max_i |e_i|_\infty,$$

where $e_i = u(z_i) - \hat{u}(z_i)$, $u(z_i)$ is the exact solution.

For simplicity, we use uniform grid points, with distance $h$ in $x$ and $y$ directions. The choice of

$h$ on finer scale is half of that on coarser scale, i.e., the number of data points in the direction of each variable on finer scale is almost twice that on coarser scale. The regularizaion parameter $\gamma$ is chosen by rules of thumb.

**Example 3.1.** *Consider the singular linear second order equation defined on a rectangular domain*

$$\nabla^2 u(x,y) = (2-\pi^2 y^2)sin(\pi x), \quad (x,y) \in [0,1]\times[0,1],$$

*with mixed Dirichlet-Neumann boundary conditions:*

$$u(0,y) = 0, \quad u(1,y) = 0,$$
$$u(x,0) = 0, \quad \frac{\partial u(x,1)}{\partial y} = 2sin(\pi x).$$

The exact solution is $u(x,y) = y^2 sin(\pi x)$. The numerical errors of multilevel algorithm are shown in Table 1 and Fig. 1.

**Table 1. Numerical errors for solving Example 1 ($a = 4h$, $\gamma = 10^7$)**

| Scale | $\lvert\mathcal{Z}_{\mathcal{D}}\rvert$ | $\lvert\mathcal{Z}_{\mathcal{B}}\rvert$ | $L_\infty$ | RMSE |
|---|---|---|---|---|
| 1 | 9 | 20 | 0.02 | $0.67\times10^{-2}$ |
| 2 | 36 | 49 | $0.10\times10^{-2}$ | $0.59\times10^{-3}$ |
| 3 | 68 | 225 | $3.46\times10^{-5}$ | $1.04\times10^{-5}$ |
| 4 | 132 | 961 | $1.60\times10^{-6}$ | $4.66\times10^{-7}$ |
| 5 | 260 | 3969 | $7.28\times10^{-8}$ | $2.10\times10^{-8}$ |

From Table 1, we observe the algorithm can perform well for this mixed boundary value problem and the numerical convergence of the approximate solution in the number of levels. As can be seen from Fig. 1, the numerical solution with good accuracy can be obtained after two scales, the accuracy is improved by at least one order of magnitude after adding a scale. But on any level of scale, the error of the numerical solution near the Neumann boundary is always the largest.

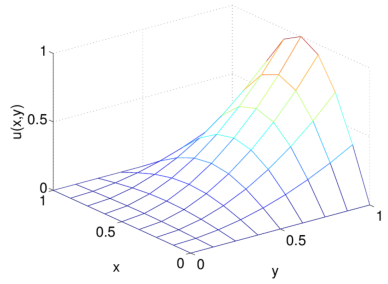
**Example 3.2.** *Consider the linear second elliptic PDE*

$$\nabla^2 u(x,y) = (2-\pi^2 y^2)sin(\pi x)$$

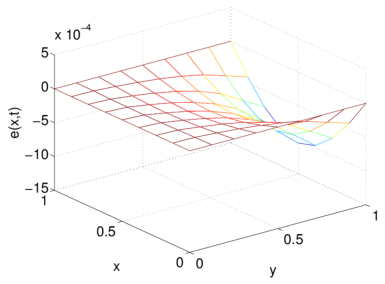
*defined on a circular domain*

$$\Sigma := \{(x,y)|x^2+y^2-1 \le 0,\ -1 \le x \le 1,\ -1 \le y \le 1\},$$

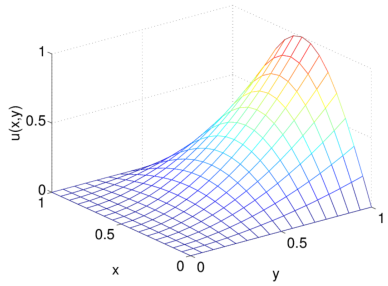
*with the Dirichlet conditions. The exact solution is given by* $u(x,y) = y^2 sin(\pi x)$.



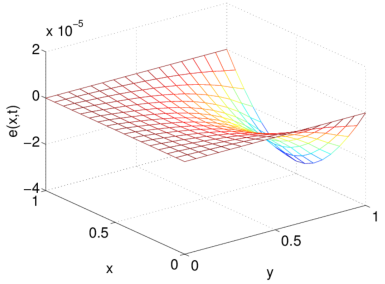

(a) Numerical solution using two scales



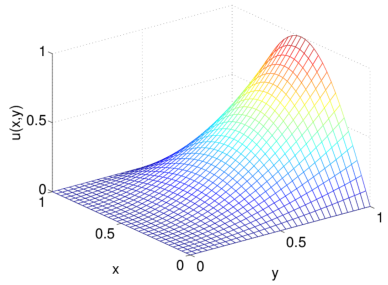

(b) Error of the solution using two scales



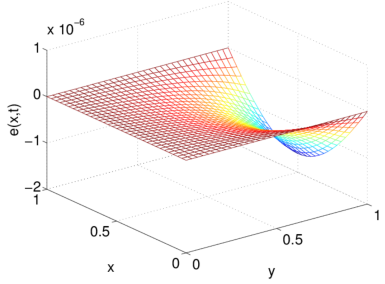

(c) Numerical solution using three scales



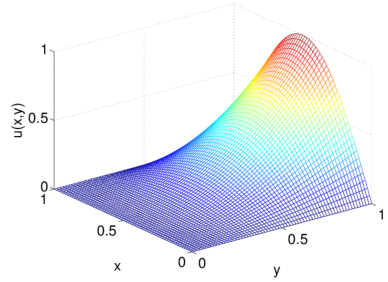

(d) Error of the solution using three scales



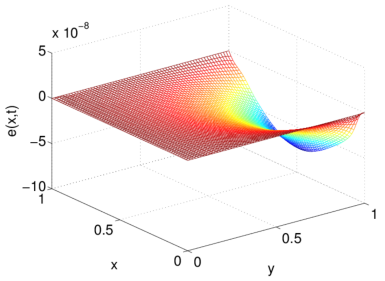

(e) Numerical solution using four scales




(f) Error of the solution using four scales

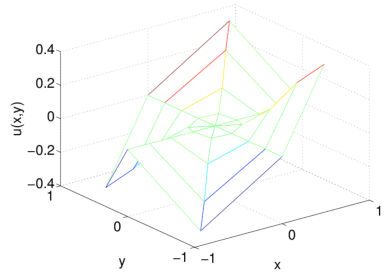



(g) Numerical solution using five scales

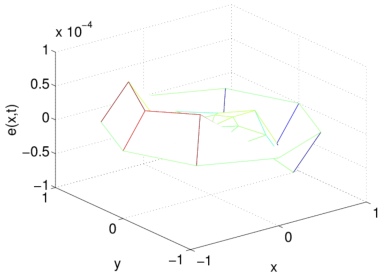



(h) Error of the solution using five scales

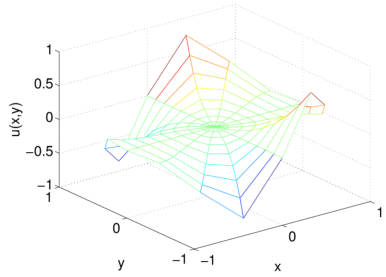
**Fig. 1. The numerical solution and errors for Example 1.**



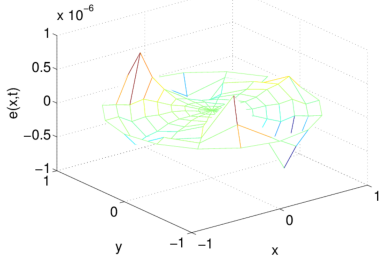

(a) Numerical solution using two scales



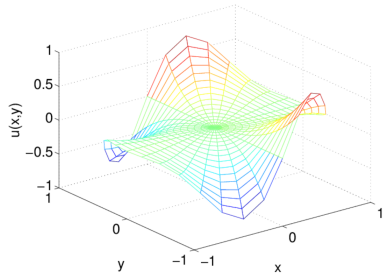

(b) Error of the solution using two scales



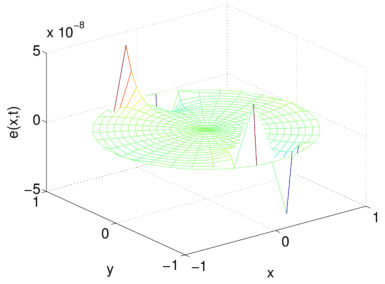

(c) Numerical solution using three scales



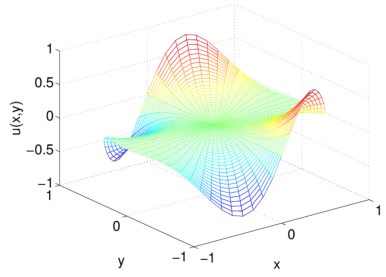

(d) Error of the solution using three scales



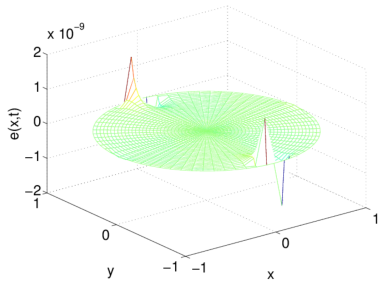

(e) Numerical solution using four scales




(f) Error of the solution using four scales




(g) Numerical solution using five scales




(h) Error of the solution using five scales

**Fig. 2. The numerical solution and errors for Example 2.**

The approximate solution and the numerical errors of multilevel algorithm for solving Example 2 are shown in Table 2 and Fig. 2. From Fig. 2, the algorithm is also effective in solving elliptic equation problems with irregular domain. Compared with example 1, the accuracy of example 2 improves faster with scale superposition, and the error is also relatively smooth in the whole domain. It can be seen that the algorithm performs better for Dirichlet boundary value problem than mixed boundary value problem.

**Table 2. Numerical errors for solving Example 2 ($a = 4h$, $\gamma = 10^7$)**

| Scale | $\lvert\mathcal{Z}_{\mathcal{D}}\rvert$ | $\lvert\mathcal{Z}_{\mathcal{B}}\rvert$ | $L_\infty$ | RMSE |
|---|---|---|---|---|
| 1 | 8 | 9 | $6.14 \times 10^{-3}$ | $2.74 \times 10^{-3}$ |
| 2 | 16 | 45 | $6.12 \times 10^{-5}$ | $2.27 \times 10^{-5}$ |
| 3 | 32 | 193 | $6.36 \times 10^{-7}$ | $1.34 \times 10^{-7}$ |
| 4 | 64 | 793 | $4.29 \times 10^{-8}$ | $4.61 \times 10^{-9}$ |
| 5 | 128 | 3205 | $1.33 \times 10^{-9}$ | $7.45 \times 10^{-11}$ |

# 4 Conclusions

We designed a multilevel LSSVM for solving differential equations based on wavelet kernel functions. The multilevel algorithm is constructed by a sequence of residual corrections and separating the computations of different levels, where different scale parameters are employed to accommodate different scales. In this algorithm, a coarse data set and a large scale parameter are chosen and the target function is interpolated in this data set to capture the large-scale variations of the target function at the first level, next, a smaller scale parameter is used to interpolate the residuals on a finer data set, capturing the finer details on the second level. The numerical tests show the efficiency of the algorithm for solving linear second order elliptic boundary value problems. The applications of the wavelet based LSSVM to evolutionary equations and more complicated nonlinear partial differential equations will be investigated in our future study.

# Competing Interests

Authors have declared that no competing interests exist.

# References

[1] Neruda R, Kudová P. Learning methods for radial basis function networks. Future Generation Computer Systems. 2005;21(7):1131-1142.

[2] Choi B, Lee JH. Comparison of generalization ability on solving differential equations using backpropagation and reformulated radial basis function networks. Neurocomputing. 2009;73(1):115-118.

[3] Lagaris IE, et al. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 1998;9(5):987-1000.

[4] Meade AJ, Fernandez AA. Solution of nonlinear ordinary differential equations by feed forward neural networks. Mathematical and Computer Modelling. 1994;20(9):19-44.

[5] Shiralashetti SC, et al. A new wavelet multigrid method for the numerical solution of elliptic type differential equations. Alexandria Engineering Journal. 2017;57:203-209.

[6] Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. Neural Processing Letter. 1999;9(3):293-300.

[7] Suykens JA, et al. Least squares support vector machines. Singapore: World Scientific Pub. Co.; 2002.

[8] Mehrkanoon S, Suykens JAK. Learning solutions to partial differential equations using LS-SVM. Neurocomputing. 2015;159:105-116.

[9] Billings SA, et al. Generalized multiscale radial basis function networks. Neural Networks. 2007;20(10):1081-1094.

[10] Nelles O. Nonlinear system identification: From classical approach to neural networks and fuzzy models. Springer-Verlag, Berlin; 2001.

[11] Farrell P, Wendland H. RBF multiscale collocation for second order elliptic boundary value problems. SIAM Journal on Numerical Analysis. 2013;51:2403-2425.

[12] Steinwart I. Learning with Hierarchical Gaussian Kernels. arXiv:1612.00824 [stat.ML]; 2016.

[13] Chui CK. An introduction to wavelets. Academic Press, Boston; 1992.

[14] Daubechies I. Ten lectures on wavelets. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania; 1992.

[15] Urban K. Wavelet methods for elliptic partial differential equations. Oxford University Press; 2009.

[16] Vapnik VN. The nature of statistical learning theory (2nd ed.). Springer-Verlag, New York; 1995.

[17] Du Z, Wang X. An improved LSSVM based on multi-scale wavelet kernel function. Control Conference. IEEE. 2012;7095-7100.

[18] Mabrouk AB, et al. A wavelet support vector machine coupled method for time series prediction. International Journal of Wavelets Multiresolution and Information Processing. 2008;6(06):851-868.

[19] Szu HH, et al. Neural network adaptive wavelets for signal representational and classification. Opt. Eng. 1992;31:1907-1916.

[20] Wang X, et al. Multi-scale least square wavelet support vector machine. 24th Chinese Control and Decision Conference (CCDC), Taiyuan. 2005;229:2879-2883.

[21] Zhang L, et al. Wavelet support vector machine. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2004;34(1):34-39.

[22] Zhang QH, Benveniste A. Wavelet networks. IEEE Trans. Neural Networks. 1992;3:889-898.

[23] Mercer J. Functions of positive and negative type, and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society of London. 1909;209:415-446.



***Peer-review history:***
*The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)*
*http://www.sciencedomain.org/review-history/26877*