

Research Article

Local Polynomial Regression Solution for Partial Differential Equations with Initial and Boundary Values

Liyun Su,¹ Tianshun Yan,¹ Yanyong Zhao,¹ and Fenglan Li²

¹ School of Mathematics and Statistics, Chongqing University of Technology, Chongqing 400054, China

² Institute of Library, Chongqing University of Technology, Chongqing 400054, China

Correspondence should be addressed to Liyun Su, cloudhopping@163.com

Received 15 April 2012; Accepted 29 July 2012

Academic Editor: Leonid Shaikhet

Copyright © 2012 Liyun Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Local polynomial regression (LPR) is applied to solve the partial differential equations (PDEs). Usually, the solutions of the problems are separation of variables and eigenfunction expansion methods, so we are rarely able to find analytical solutions. Consequently, we must try to find numerical solutions. In this paper, two test problems are considered for the numerical illustration of the method. Comparisons are made between the exact solutions and the results of the LPR. The results of applying this theory to the PDEs reveal that LPR method possesses very high accuracy, adaptability, and efficiency; more importantly, numerical illustrations indicate that the new method is much more efficient than B-splines and AGE methods derived for the same purpose.

1. Introduction

There are some effective and convenient numerical methods for partial differential equations problems with initial and boundary values; for example, the radial basis functions are used to solve two-dimensional sine-Gordon equation in [1], a family of second-order methods are used to solve variable coefficient fourth-order parabolic partial differential equations in [2], and fifth-degree B-spline solution is available for a fourth-order parabolic partial differential equations in [3]. Recently, H. Caglar and N. Caglar [4] have used local polynomial regression (LPR) method for the numerical solution of fifth-order boundary value problems [4]. Moreover, they manage to solve linear Fredholm and Volterra integral equations by using LPR method [5]. Consequently, we can consider LPR method is applied to some PDEs with initial and boundary values, and numerical results demonstrate that local polynomial

fitting method is more accurate, simple, and efficient. First, we consider the numerical approximation for the simple nonhomogeneous PDE with initial values with the use of local polynomial regression:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} &= g(x, t), \\ u|_{t=0} &= \varphi(x), \quad \left. \frac{\partial u}{\partial t} \right|_{t=0} = \psi(x). \end{aligned} \quad (1.1)$$

2. Bivariate Local Polynomial Regression

Bivariate local polynomial regression is an attractive method both from theoretical and practical point of view. Multivariate local polynomial method has a small mean squared error compared with the Nadaraya-Watson estimator which leads to an undesirable form of the bias and the Gasser-Muller estimator which has to pay a price in variance when dealing with a random design model. Multivariate local polynomial fitting also has other advantages. The method adapts to various types of designs such as random and fixed designs and highly clustered and nearly uniform designs. Furthermore, there is an absence of boundary effects: the bias at the boundary stays automatically of the same order as the interior, without the use of specific boundary kernels. The local polynomial approximation approach is appealing on general scientific grounds; the least squares principle to be applied opens the way to a wealth of statistical knowledge and thus easy generalizations. All the above-mentioned assertions or advantages can be found in literatures [6–10]. The basic idea of multivariate local polynomial regression is also proposed in [11–14]. In this section, we briefly outline the idea of the extension of bivariate local polynomial fitting to bivariate regression.

Suppose that the state vector at point T is X_T :

$$y_{i,T} = f_i(X_T). \quad (2.1)$$

Our purpose is to obtain the estimation $\hat{y}_{i,T} = \hat{f}_i(X_T)$. In this paper, we use the d th order multivariate local polynomial $f_i(X)$ to predict the value of the fixed-point X_T . The polynomial function can be described as

$$f_i(X) \approx \sum_{0 \leq |j| \leq d} \frac{1}{j!} D^{(j)} f_i(X_T) (X - X_T)^j = \sum_{0 \leq |j| \leq d} b_j(X_T) (X - X_T)^j, \quad (2.2)$$

where

$$\begin{aligned} j &= (j_1, j_2), \quad j! = j_1! j_2!, \quad |j| = \sum_{l=1}^2 j_l, \\ \sum_{0 \leq |j| \leq d} &= \sum_{|j|=0}^d \left(\sum_{j_1=0}^{|j|} \sum_{j_2=0}^{|j|} \right), \quad X^j = x_1^{j_1} x_2^{j_2}. \end{aligned} \quad (2.3)$$

d is the order of the expansion:

$$D^{(j)} f_i(X_T) = \frac{\partial^{|j|} f_i(X)}{\partial x_1^{j_1} \partial x_2^{j_2} \cdots \partial x_m^{j_m}} \Big|_{X = X_T}, \quad (2.4)$$

$$b_j(X_T) = \frac{1}{j!} D^{(j)} f_i(X_T).$$

In the bivariate regression method, the change of X_T on the attractor is assumed to be the same as those of nearby points, X_{T_a} ($a = 1, 2, \dots, A$) according to the distance order. Using A pairs of (X_{T_a}, y_{i,T_a}) , for which the values are already known, the coefficients of f_i are determined by minimizing

$$\sum_{a=1}^A \left[y_{i,T_a} - \sum_{0 \leq |j| \leq d} b_j(X_T) (X_{T_a} - X_T)^j \right]^2 \cdot K_H(X_{T_a} - X_T). \quad (2.5)$$

For the weighted least squared problem, when $X^T W X$ is inverse, the solution can be described by

$$\hat{B} = (X^T W X)^{-1} X^T W Y, \quad (2.6)$$

where

$$Y = (y_{i,T_1}, y_{i,T_2}, \dots, y_{i,T_A})^T, \quad (2.7)$$

$$\hat{B} = (b_0(X_T), b_1(X_T), \dots, b_d(X_T))^T,$$

$$W = \text{diag}\{K_H(X_{T_1} - X_T), K_H(X_{T_2} - X_T), \dots, K_H(X_{T_A} - X_T)\},$$

and X is the $A \times S$ ($S = \sum_{0 \leq |j| \leq d} |j|/j!$),

$$X = \begin{pmatrix} 1 & (X_{T_1} - X_T)^1 & \cdots & (X_{T_1} - X_T)^d \\ 1 & (X_{T_2} - X_T)^1 & \cdots & (X_{T_2} - X_T)^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (X_{T_A} - X_T)^1 & \cdots & (X_{T_A} - X_T)^d \end{pmatrix}. \quad (2.8)$$

Then we can get the estimation $\hat{y}_{i,T} = \hat{f}_i(X_T)$:

$$\hat{y}_{i,T} = \hat{f}_i(X_T) = E_i (X^T W X)^{-1} X^T W Y, \quad (2.9)$$

where $E_i = (1, 0, 0, \dots, 0)_{1 \times S}$. There are several important issues about the bandwidth, the order of multivariate local polynomial function, and the kernel function which have to be discussed.

2.1. Parameters Estimations and Selections

There are many of the embedding dimensions algorithms [15, 16]. In univariate series $\{y_{i,n}\}_{n=1}^N, i = 1, 2, \dots, M$, a popular method that is used for finding the embedding dimensions m_i is the so-called false nearest-neighbor method [17, 18]. Here, we apply this method to the bivariate case.

For the multivariate local polynomial estimator, there are three important problems which have significant influence on the prediction accuracy and computational complexity. First of all, there is the choice of the bandwidth matrix, which plays a rather crucial role. The bandwidth matrix H is taken to be a diagonal matrix. For simplification, the bandwidth matrix is designed into $H = hI_2$, where I_2 is a unit matrix of two orders. In theory, there exists an optimal bandwidth h_{opt} in the meaning of mean squared error, such that

$$h_{\text{opt}} = \arg \min_h \int \left(f_i(x) - \hat{f}_i(x) \right)^2 dx. \quad (2.10)$$

But the optimal bandwidth cannot be solved directly. So we discuss how to get the asymptotically optimal bandwidth. We make use of searching method to select bandwidth. When the bandwidth h varies from small to big, compared with the values of the objective function, we can choose the smallest bandwidth to ensure the minimum value of the objective function. So the smallest bandwidth is the optimal bandwidth.

Given $h_l = C_l h_{\text{min}}$, where h_{min} is minimum, C_l is efficient of expansion. We search the bandwidth h to ensure the objective function value of minimum in district $[h_{\text{min}}, h_{\text{max}}]$, where the objective function stands for the mean square error (MSE).

First, we assume $h = h_{\text{min}}$ and then increase h by making use of efficient of expansion C_l . When $h > h_{\text{max}}$ stops, the minimum h is the optimal bandwidth if $E_{\text{MS}}(h)$ obtains the minimum.. $E_{\text{MS}}(h)$ can be taken place of

$$e_{\text{MS}} = \frac{1}{I} \sum_{i=1}^I \left(f_i(x) - \hat{f}_i(x) \right)^2, \quad (2.11)$$

where I is the number predicted. In this paper, we choose

$$h_{\text{min}} = \frac{2h_m}{n}, \quad h_{\text{max}} = \frac{h_m}{2}, \quad (2.12)$$

where $h_m = \max \|X_i - X_j\|, C_l = 1.1$. Compared to other methods, this method is more convenient.

In order to get closer to the ideal optimal bandwidth, we search once again by narrowing the interval on the basis of the previous searching process. Suppose j is the bandwidth which makes $C_l^j h_{\text{min}}$ optimal in the above searching process. Now, small interval $[C_l^{j-1} h_{\text{min}}, C_l^{j+1} h_{\text{min}}]$ is divided into n equal intervals again. Suppose

$$h_i = h_{\text{min}} C_l^{j-1} \left(1 + \frac{i}{n} (C_l^2 - 1) \right) \quad (i = 1, \dots, n-1). \quad (2.13)$$

Among $n-1$ bandwidth, the bandwidth that makes e_{MS} minimized is the optimal bandwidth.

Another issue in multivariate local polynomial fitting is the choice of the order of the polynomial. Since the modeling bias is primarily controlled by the bandwidth, this issue is less crucial, however. For a given bandwidth h , a large value of d would expectedly reduce the modeling bias but would cause a large variance and a considerable computational cost. Since the bandwidth is used to control the modeling complexity, and due to the sparsity of local data in multidimensional space, a higher-order polynomial is rarely used. So, we apply the local quadratic regression to fit the model (i.e., $p = 2, 3$). The third issue is the selection of the kernel function. In this paper, we choose the optimal spherical *Epanechnikov* kernel function [6, 7] which minimizes the asymptotic mean square error (MSE) of the resulting multivariate local polynomial estimators, as our kernel function.

3. LPR Solutions for PDE

For the following nonhomogeneous PDE with boundary values:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} &= g(x, t), \\ u|_{t=0} &= \varphi(x), \quad \frac{\partial u}{\partial t} \Big|_{t=0} = \psi(x), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1. \end{aligned} \quad (3.1)$$

Suppose $X_T = (x_T, t_T)$, so our purpose is to obtain estimation $\hat{y}_{i,T}$ given n^2 pairs of points $X_{T_i} = (x_{T_i}, t_{T_i})$, where $\hat{y}_{i,T} = \hat{f}_i(X_T)$, $x_{T_1} = 0 < x_{T_2} < \dots < x_{T_n} = 1$, $t_{T_1} = 0 < t_{T_2} < \dots < t_{T_n} = 1$; then we can get

$$u_{tt}(X_T) - u_{xx}(X_T) \approx g_i(X_T), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \quad (3.2)$$

that is described by

$$\begin{aligned} & \left[\beta_0 + \beta_1(x - x_0) + \beta_2(t - t_0) + \beta_3(x - x_0)^2 + \beta_4(x - x_0)(t - t_0) + \beta_5(t - t_0)^2 + \dots \right]_{tt} \\ & - \left[\beta_0 + \beta_1(x - x_0) + \beta_2(t - t_0) + \beta_3(x - x_0)^2 + \beta_4(x - x_0)(t - t_0) + \beta_5(t - t_0)^2 + \dots \right]_{xx} \\ & \approx g_i(X_T). \end{aligned} \quad (3.3)$$

Consequently, the corresponding minimization function is

$$\begin{aligned} & \left[\sum_{T_1=1}^n [\varphi(x_{T_1}, 0) - u(x_{T_1}, 0)]^2 \cdot K_H(X_{T_1} - X_T) + \sum_{T_2=1}^n [\psi(x_{T_2}, 0) - u_t(x_{T_2}, 0)]^2 \right] \\ & \cdot K_H(X_{T_2} - X_T) + \left[\sum_{i=2}^{n-1} \sum_{j=1}^n \left[g(x_{T_i}, t_{T_j}) - [u_{tt}(x_{T_i}, t_{T_j}) - u_{xx}(x_{T_i}, t_{T_j})] \right]^2 \cdot K_H(X_{T_i} - X_T) \right]. \end{aligned} \quad (3.4)$$

In order to find expression of X , we need to find the elements of X at first.

Since binary function's p -order Taylor expansion has $(p^2 + 3p + 2)/2$ items, then we can get expressions (3.5), (3.6), and (3.7):

$$X = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & \cdots \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{n5} & a_{n6} & \cdots \\ d_{(n+1),1} & d_{(n+1),2} & d_{(n+1),3} & d_{(n+1),4} & d_{(n+1),5} & d_{(n+1),6} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ d_{n(n-1),1} & d_{n(n-1),2} & d_{n(n-1),3} & d_{n(n-1),4} & d_{n(n-1),5} & d_{n(n-1),6} & \cdots \\ g_{[n(n-1)+1],1} & g_{[n(n-1)+1],2} & g_{[n(n-1)+1],3} & g_{[n(n-1)+1],4} & g_{[n(n-1)+1],5} & g_{[n(n-1)+1],6} & \cdots \\ g_{[n(n-1)+2],1} & g_{[n(n-1)+2],2} & g_{[n(n-1)+2],3} & g_{[n(n-1)+2],4} & g_{[n(n-1)+2],5} & g_{[n(n-1)+2],6} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ g_{n^2,1} & g_{n^2,2} & g_{n^2,3} & g_{n^2,4} & g_{n^2,5} & g_{n^2,6} & \cdots \end{bmatrix}, \quad (3.5)$$

where X is $n^2 \times (p+1)(p+2)/2$ matrix:

$$\begin{aligned} i = 1, 2, \dots, n, \quad j = 1, \quad a_{ij} &= 1 \\ j = 2, \quad a_{ij} &= (x_{T_i} - x_T) \\ j = 3, \quad a_{ij} &= (t_{T_i} - t_T) \\ j = 4, \quad a_{ij} &= (x_{T_i} - x_T)^2 \\ j = 5, \quad a_{ij} &= (x_{T_i} - x_T)(t_{T_i} - t_T) \\ j = 6, \quad a_{ij} &= (t_{T_i} - t_T)^2, \end{aligned} \quad (3.6)$$

$$\begin{aligned} i = n+1, n+2, \dots, 2n, \dots, n(n-1), \quad j = 1, 2, 3, 5, \quad d_{ij} &= 0 \\ j = 4, 6 \quad d_{ij} &= 1, \end{aligned}$$

$$\begin{aligned} i = n(n-1)+1, n(n-2)+2, \dots, n^2, \quad j = 1, 2, 4, \quad g_{ij} &= 0 \\ j = 3 \quad g_{ij} &= 1 \end{aligned}$$

$$k = 1, 2, \dots, n \quad j = 5 \quad g_{ij} = (x_{T_k} - x_T)$$

$$k = 1, 2, \dots, n \quad j = 6 \quad g_{ij} = 2(x_{T_k} - x_T),$$

$$\begin{aligned} Y^T = [\varphi(x_{T_1}) \quad \varphi(x_{T_2}) \quad \cdots \quad \varphi(x_{T_n}) \quad g(x_{T_1}, t_{T_2}) \quad g(x_{T_2}, t_{T_2}) \quad \cdots \quad g(x_{T_n}, t_{T_2}) \\ \cdots \quad g(x_{T_1}, t_{T_{n-1}}) \quad g(x_{T_2}, t_{T_{n-1}}) \quad \cdots \quad g(x_{T_n}, t_{T_{n-1}}) \quad \psi(x_{T_1}) \quad \psi(x_{T_2}) \quad \cdots \quad \psi(x_{T_n})]_{1 \times n^2}. \end{aligned} \quad (3.7)$$

Substitute (3.6) and (3.7) into

$$\hat{B} = (X^T W X)^{-1} X^T W Y. \quad (3.8)$$

Now, we can get the estimation $\hat{y}_{i,T} = \hat{f}_i(X_T)$, ($i = 1, 2, \dots, n$),

$$\hat{y}_{i,T} = \hat{f}_i(X_T) = E_i \left(X^T W X \right)^{-1} X^T W Y, \quad (3.9)$$

where $E_i = (1, 0, 0, \dots, 0)_{1 \times S}$ ($S = \sum_{0 \leq |j| \leq d} |j|/j!$).

4. Numerical Illustrations and Discussions

In this section we consider the numerical results obtained by the schemes discussed previously by applying them to the following second-order and fourth-order initial boundary value problems. All computations are carried out using MATLAB 7.0.

Furthermore, in order to evaluate the accuracy and effectiveness, we apply the following indices, namely, the mean squared prediction error (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left(f_i(x) - \hat{f}_i(x) \right)^2 \quad (4.1)$$

and the absolute error

$$\text{AE}_i = f(x_i) - \hat{f}(x_i). \quad (4.2)$$

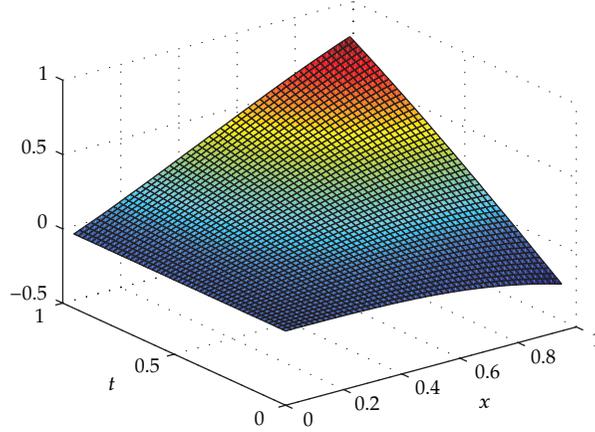
Example 4.1. First, we consider the following second-order nonhomogeneous PDEs:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} &= t \sin x \\ u|_{t=0} &= 0, \quad \left. \frac{\partial u}{\partial t} \right|_{t=0} = \sin x. \end{aligned} \quad (4.3)$$

The exact solution for problems (4.3) is $u(x, t) = t \sin x$. We solve Example 4.1 with $n = 20, 30, 50$ by choosing $p = 3, 4$ and various values of parameters H_i presented in Table 1. The errors in the solutions are computed by our method (3.9). In Table 1, $p = 3, 4$. Given $n = 20, p = 3, 4$, it is found that the magnitude of MSE is between 10 to the power of -3 and 10 to the power of -4 . Given $n = 30, p = 3, 4$, it can be seen that the magnitude of MSE is between 10 to the power of -4 and 10 to the power of -7 . However, by setting up $n = 50, p = 3, 4$, it is obvious that the magnitude of MSE is between 10 to the power of -5 and 10 to the power of -8 . We conclude that MSE decreases with the increase of the value of n the value of order p has no much influence on MSE for a large value of β . Consider parameter H ; specifically, MSEs get up to minimum for $p = 3, n = 20$ when $H_3 = 1/25$, for $p = 3, n = 30$ when $H_4 = 3/100$, for $p = 3, n = 50$ when $H_5 = 1/40$. We can find that the optimal bandwidth H locates in district $[0.025I_2, 0.04I_2]$ by using method (2.12)-(2.13) for $n = 20, 30, 50$. For $p = 4$, there exists the same situation with $p = 3$. Here, I_2 is a unit matrix of two orders. The fitting figure is depicted in Figure 1 using matlab 7.0.

Table 1: Mean square errors (MSEs) with different values of p, H given $n = 20, 30, 50$ in Example 4.1, I_2 : unit matrix of two orders.

Parameters (p, H_i)	MSE ($n = 20$)	MSE ($n = 30$)	MSE ($n = 50$)
$p = 3, H_1 = (11/100)I_2$	8.5223×10^{-3}	6.5223×10^{-4}	6.3252×10^{-5}
$p = 3, H_2 = (2/25)I_2$	3.0309×10^{-3}	4.2541×10^{-4}	9.7206×10^{-6}
$p = 3, H_3 = (1/25)I_2$	1.1268×10^{-4}	8.3418×10^{-5}	6.0211×10^{-6}
$p = 3, H_4 = (3/100)I_2$	5.3082×10^{-4}	5.2087×10^{-5}	1.1528×10^{-6}
$p = 3, H_5 = (1/40)I_2$	6.9123×10^{-4}	5.6021×10^{-5}	3.9107×10^{-7}
$p = 3, H_6 = (1/75)I_2$	3.3082×10^{-3}	2.8339×10^{-5}	1.8815×10^{-6}
$p = 4, H_1 = (11/100)I_2$	4.1004×10^{-4}	6.6033×10^{-6}	5.8311×10^{-7}
$p = 4, H_2 = (2/25)I_2$	1.8791×10^{-4}	2.2416×10^{-6}	3.4525×10^{-7}
$p = 4, H_3 = (1/25)I_2$	6.6568×10^{-5}	9.5066×10^{-7}	3.0373×10^{-7}
$p = 4, H_4 = (3/100)I_2$	9.0893×10^{-5}	7.3137×10^{-6}	9.7815×10^{-8}
$p = 4, H_5 = (1/40)I_2$	5.9803×10^{-4}	5.6401×10^{-6}	4.1266×10^{-8}
$p = 4, H_6 = (1/75)I_2$	5.7911×10^{-4}	7.3452×10^{-6}	2.3098×10^{-7}

**Figure 1:** LPR of Example 4.1.

Example 4.2. Next we consider the following fourth-order nonhomogeneous PDEs:

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^4 u}{\partial x^4} = (\pi^4 - 1) \sin \pi x \cos t, \quad 0 \leq x \leq 1, \quad t > 0, \quad (4.4)$$

with the initial conditions,

$$u(x, 0) = \sin \pi x, \quad u_t(x, 0) = 0, \quad 0 \leq x \leq 1, \quad (4.5)$$

and the boundary conditions at $x = 0$ and $x = 1$ of the form

$$u(0, t) = u(1, t) = \frac{\partial^2 u}{\partial x^2}(0, t) = \frac{\partial^2 u}{\partial x^2}(1, t) = 0, \quad t \geq 0. \quad (4.6)$$

Table 2: Mean square errors (MSEs) with different values of p, H given $n = 30$ and absolute errors (AE) given $n = 30, 50$ in Example 4.2, I_2 : unit matrix of two orders.

Parameters (h, H_i)	MSE ($n = 30$)	AE ($x = 0.10, n = 30$)	AE ($x = 0.20, n = 50$)
$p = 4, H_1 = (11/100)I_2,$	2.3109×10^{-4}	4.7882×10^{-3}	9.4388×10^{-5}
$p = 4, H_2 = (2/25)I_2,$	4.1493×10^{-4}	3.1431×10^{-3}	4.6412×10^{-5}
$p = 4, H_3 = (1/25)I_2,$	3.8981×10^{-5}	6.9083×10^{-4}	1.0603×10^{-5}
$p = 4, H_4 = (3/100)I_2,$	9.2391×10^{-6}	1.1425×10^{-4}	7.5229×10^{-6}
$p = 4, H_5 = (1/40)I_2,$	9.0117×10^{-6}	6.8858×10^{-4}	9.1029×10^{-7}
$p = 4, H_6 = (3/100)I_2,$	7.3249×10^{-6}	9.4551×10^{-5}	2.1002×10^{-6}
$p = 5, H_1 = (11/100)I_2,$	7.0113×10^{-7}	8.9211×10^{-5}	6.8564×10^{-6}
$p = 5, H_2 = (2/25)I_2,$	4.5881×10^{-7}	7.8423×10^{-5}	5.0705×10^{-6}
$p = 5, H_3 = (1/25)I_2,$	2.1062×10^{-7}	2.6065×10^{-5}	9.6552×10^{-7}
$p = 5, H_4 = (3/100)I_2,$	9.7922×10^{-8}	5.8232×10^{-6}	8.2302×10^{-7}
$p = 5, H_5 = (1/40)I_2,$	3.8717×10^{-6}	4.2410×10^{-5}	3.0671×10^{-7}
$p = 5, H_6 = (1/75)I_2,$	8.9821×10^{-6}	7.9102×10^{-5}	6.3409×10^{-6}
Aziz et al. [19] (p, q, s, σ, h)		$x = 0.10, \text{steps} = 10$	$x = 0.20, \text{steps} = 16$
(0, 0, 1, 1/4, 1/20)	—	1.5000×10^{-4}	5.1000×10^{-5}
(0, 1/6, 2/3, 1/4, 1/20)	—	1.8000×10^{-5}	8.0000×10^{-6}
(1/144, 5/36, 17/24, 1/4, 1/20)	—	1.8000×10^{-5}	7.9000×10^{-6}
Evans and Yousif [20]	—	2.2000×10^{-4}	4.1000×10^{-4}
		2.5000×10^{-5}	4.7000×10^{-5}

The exact solution for problems (4.4), (4.5), and (4.6) is $u(x, t) = \sin \pi x \cos t$. This problem has been solved by several authors [19, 20]. Here, we try to solve Example 4.2 with $n = 30, 50$ by choosing $p = 3, 4$ and various values of parameters H presented in Table 2. The errors in the solutions are computed by our method (3.9); the splines method for three kinds of values of p, q, s, σ, h in [19], and the AGE method for two kinds of values of its parameters in [20] have been presented in Table 2. In Table 1, $p = 3, 4$. Given $n = 30$, it is found that the magnitude of MSE is between 10 to the power of -4 and 10 to the power of -6 for $p = 3$, between 10 to the power of -6 and 10 to the power of -8 for $p = 4$. We can see the value of order p has a little influence on MSE. Consider parameter H ; specifically, MSEs get up to minimum for $p = 3, n = 20$ when $H_3 = 1/25$, for $p = 3, n = 30$ when $H_4 = 3/100$, for $p = 3, n = 50$ when $H_5 = 1/40$. Similar to Example 4.1, we can also acknowledge the optimal bandwidth H exists in district $[0.025I_2, 0.03I_2]$ by using method (2.12)-(2.13), for $n = 30$ and $n = 50$ given $p = 4, 5$. Furthermore, we have tabulated the absolute errors (AEs) at $x = 0.10, 0.20$ for different combination of parameters p, H for $n = 30$ and $n = 50$, respectively. Given $n = 30, p = 4, 5$, the magnitude of AE is between 10 to the power of -3 and 10 to the power of -6 at point $x = 0.1$, and given $n = 50, p = 4, 5$, the magnitude of AE is between 10 to the power of -5 and 10 to the power of -7 at point $x = 0.2$. Here, I_2 is a unit matrix of two orders. The fitting figure is also illustrated in Figure 2 using matlab 7.0.

5. Conclusions

In this paper, LPR method can be applied for the numerical solution of some kinds of PDEs. We can also see that LPR method has been exploited to solve fifth-order boundary value

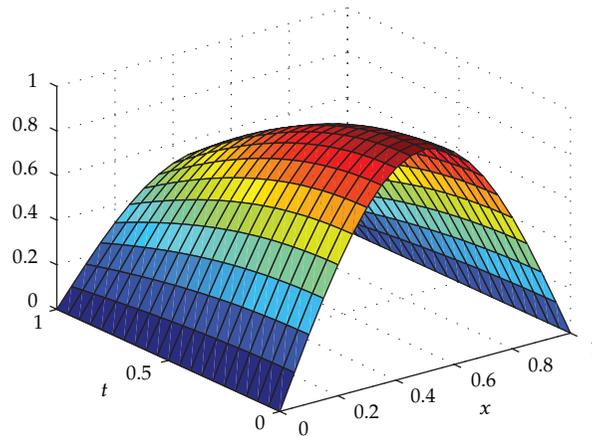


Figure 2: LPR of Example 4.2.

problems [4] and Fredholm and Volterra integral equations [5] whose maximum absolute errors are very small and calculation processes are simple and feasible. Compared with the splines method [19] and the AGE method [20], LPR method converges to solutions with fewer number of nodes, and the maximum absolute errors are a little smaller. Moreover, it is more flexible to resolve problems just only adjusting parameters h, p . However, LPR methods have shortcomings which need more computation time compared with splines method [19] for the same problems, which can be tried to discuss and resolve afterward. In any case, it is more important that we can conclude LPR solution is a powerful tool for numerical solutions of differential equations with initial and boundary values.

Acknowledgment

This work was supported by Natural Science Foundation Projects of CQ CSTC of China (CSTC2010BB2310, CSTC2011jjA40033, CSTC2012jjA00037) and Chongqing CMEC Foundations of China (KJ080614, KJ120829).

References

- [1] M. Dehghan and A. Shokri, "A numerical method for solution of the two-dimensional sine-Gordon equation using the radial basis functions," *Mathematics and Computers in Simulation*, vol. 79, no. 3, pp. 700–715, 2008.
- [2] A. Q. M. Khaliq and E. H. Twizell, "A family of second order methods for variable coefficient fourth order parabolic partial differential equations," *International Journal of Computer Mathematics*, vol. 23, pp. 63–76, 1987.
- [3] H. Caglar and N. Caglar, "Fifth-degree B-spline solution for a fourth-order parabolic partial differential equations," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 597–603, 2008.
- [4] H. Caglar and N. Caglar, "Solution of fifth order boundary value problems by using local polynomial regression," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 952–956, 2007.
- [5] H. Caglar and N. Caglar, "Numerical solution of integral equations by using local polynomial regression," *Journal of Computational Analysis and Applications*, vol. 10, no. 2, pp. 187–195, 2008.
- [6] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications*, vol. 66, Chapman & Hall, London, UK, 1996.

- [7] J. Fan and Q. Yao, *Nonlinear Time Series*, Springer, New York, NY, USA, 2003.
- [8] J. Fan and I. Gijbels, "Adaptive order polynomial fitting: bandwidth robustification and bias reduction," *Journal of Computational and Graphical Statistics*, vol. 4, no. 3, pp. 213–227, 1995.
- [9] J. Fan and I. Gijbels, "Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation," *Journal of the Royal Statistical Society B*, vol. 57, no. 2, pp. 371–394, 1995.
- [10] J. Fan, N. E. Heckman, and M. P. Wand, "Local polynomial kernel regression for generalized linear models and quasi-likelihood functions," *Journal of the American Statistical Association*, vol. 90, no. 429, pp. 141–150, 1995.
- [11] L. Su and F. Li, "Deconvolution of defocused image with multivariate local polynomial regression and iterative wiener filtering in DWT domain," *Mathematical Problems in Engineering*, vol. 2010, Article ID 605241, 14 pages, 2010.
- [12] L. Su, Y. Zhao, and T. Yan, "Two-stage method based on local polynomial fitting for a linear heteroscedastic re-gression model and its application in economics," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 696927, 17 pages, 2012.
- [13] L.-Y. Su, "Prediction of multivariate chaotic time series with local polynomial fitting," *Computers & Mathematics with Applications*, vol. 59, no. 2, pp. 737–744, 2010.
- [14] L. Su, Y. Ma, and J. Li, "Application of local polynomial estimation in suppressing strong chaotic noise," *Chinese Physics B*, vol. 21, no. 2, Article ID 020508, 2012.
- [15] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403–3411, 1992.
- [16] S. Boccaletti, D. L. Valladares, L. M. Pecora, H. P. Geffert, and T. Carroll, "Reconstructing embedding spaces of coupled dynamical systems from multivariate data," *Physical Review E*, vol. 65, no. 3, Article ID 035204, pp. 1–4, 2002.
- [17] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, vol. 7, Cambridge University Press, Cambridge, UK, 1997.
- [18] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, no. 2, pp. 1134–1140, 1986.
- [19] T. Aziz, A. Khan, and J. Rashidinia, "Spline methods for the solution of fourth-order parabolic partial differential equations," *Applied Mathematics and Computation*, vol. 167, no. 1, pp. 153–166, 2005.
- [20] D. J. Evans and W. S. Yousif, "A note on solving the fourth order parabolic equation by the AGE method," *International Journal of Computer Mathematics*, vol. 40, pp. 93–97, 1991.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

