

## Research Article

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

Liyun Su,<sup>1</sup> Tianshun Yan,<sup>1</sup> Yanyong Zhao,<sup>1</sup> and Fenglan Li<sup>2</sup>

<sup>1</sup> School of Mathematics and Statistics, Chongqing University of Technology, Chongqing 400054, China

<sup>2</sup> Library of Chongqing University of Technology, Chongqing 400054, China

Correspondence should be addressed to Liyun Su; [cloudhopping@163.com](mailto:cloudhopping@163.com)

Received 13 March 2013; Accepted 7 May 2013

Academic Editor: Yuji Liu

Copyright © 2013 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.

Numerical solutions of the linear differential boundary issues are obtained by using a local polynomial estimator method with kernel smoothing. To achieve this, a combination of a local polynomial-based method and its differential form has been used. The computed results with the use of this technique have been compared with the exact solution and other existing methods to show the required accuracy of it. The effectiveness of this method is verified by three illustrative examples. The presented method is seen to be a very reliable alternative method to some existing techniques for such realistic problems. Numerical results show that the solution of this method is more accurate than that of other methods.

## 1. Introduction

Linear ordinary differential equation boundary value problems are very common theoretical and practical issues. Some solutions are put forward in [1–6]. Most solutions usually include finite difference method and shooting method. The shooting method is the most widely used. Both single and multishooting methods: (1) select the initial value of needing to be solved; (2) use the differential equation initial value algorithm to compute step by step and then get to the end of boundary value; (3) compute the error function between boundary value to compute and real boundary value then readjust the initial value of needing to be solved until the error is up to a certain level of accuracy. In general, this computing process is more tedious. This text makes use of local polynomial estimation to solve the boundary value issues of homogeneous and nonhomogeneous linear differential equations and compare it with the method in [4].

The linear differential equation below has a unique solution:

$$\begin{aligned}
 y''(x) + p(x)y'(x) + q(x)y(x) &= f(x), \\
 y(a) &= y_0, \quad y(b) = y_1.
 \end{aligned}
 \tag{1}$$

Among them,  $y_0, y_1$  are constant, and the function  $p(x), q(x), f(x)$  is continuous in  $[a, b]$ .

The presented method is useful for obtaining numerical approximations of differential equations, and it is also quite straightforward to write corresponding codes in any programming languages. Also, roundoff errors and necessity for large computer memory are not faced in this method. The computed results obtained by this way have been compared with the exact solution to show the required accuracy of it and the existing methods. Furthermore, the current method is of a general nature and can therefore be used for solving some partial differential equations arising in various areas.

## 2. Local Polynomial Estimator for Differential Equations

In this paper, we take advantage of the numerical estimation with the use of local polynomial regression to solve the boundary value problem. The basic idea of this method is first proposed in [5]. First, we introduce the mathematical thoughts of local polynomial regression. This idea was mentioned in [7–10]. Since the form of regression function is not specified, so the data points with long distance from  $x_0$  provide little information to  $y(x_0)$ . Therefore, we can only use

the local data points around  $x_0$ . We suppose that  $y(x)$  has  $p+1$  derivative at  $x_0$ , by the Taylor expansion, for point  $x$ , located in the neighborhood of this point  $x_0$ ; we can use the  $p$ -order multivariate polynomials to locally approximate  $y(x)$  and the surrounding local point of  $x_0$ ; we model  $y(x)$  as

$$y(x) \approx \sum_{j=0}^p \beta_j (x - x_0)^j, \quad (2)$$

where parameter  $\beta_j$  depends on  $x_0$ , so-called local parameter. Obviously, the local parameter  $\beta_j = y^{(j)}(x_0)/j!$  fit the local model with local data, and it can be minimized:

$$\sum_{i=1}^n \left[ Y_i - \sum_{j=0}^p \beta_j (X_i - x_0)^j \right]^2 K_h(X_i - x), \quad (3)$$

where  $h$  controls the size of the bandwidth of local area. Using matrix notation to represent the local polynomial regression is more convenient. Below is the design matrix corresponding to (4) with  $X$  and  $Y$ :

$$X = \begin{pmatrix} 1 & X_1 - x_0 & \cdots & (X_1 - x_0)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n - x_0 & \cdots & (X_n - x_0)^p \end{pmatrix}, \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}. \quad (4)$$

The weighted least squares problem (4) can be written as

$$\min (Y - X\beta)^T W (Y - X\beta), \quad (5)$$

here,

$$W = \text{diag}(K_h(X_1 - x_0), \dots, K_h(X_n - x_0)), \quad (6)$$

so the solution vector is

$$\beta = (X^T W X)^{-1} X^T W Y. \quad (7)$$

For the initial value problem (1) and (3), we choose the best quadratic kernel function (Fan and Yao have proved in [10]):

$$K_h(u) = \begin{cases} \frac{3}{4}(1 - u^2)_+, & \text{if } |u| \leq 1; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

### 3. Local Polynomial Estimator

Consider the following:

$$\begin{aligned} y''(x) + p(x)y'(x) + q(x) &= f(x), \quad a \leq x \leq b, \\ y(a) &= y_a, \quad y(b) = y_b. \end{aligned} \quad (9)$$

Here, we suppose that  $X_1 = a < X_2 < \dots < X_n = b$ , then we obtain the following relation:

$$\begin{aligned} \left( \sum_{j=0}^p \beta_j (x - x_0)^j \right)'' + \sum_{j=1}^p p(x) j \beta_j (x - x_0)^{j-1} \\ + \sum_{j=0}^p q(x) \beta_j (x - x_0)^j \approx f(x), \quad a \leq x \leq b. \end{aligned} \quad (10)$$

Then, the corresponding minimization function is

$$\begin{aligned} & \left\{ \left[ y_a - \sum_{j=0}^p \beta_j (X_1 - x_0)^j \right]^2 \right. \\ & + \sum_{i=2}^{n-1} \left[ f(X_i) - \sum_{j=2}^p j(j-1) \beta_j (X_i - x_0)^{j-2} \right. \\ & \quad \left. - \sum_{j=1}^p p(x) j \beta_j (X_i - x_0)^{j-1} \right. \\ & \quad \left. - \sum_{j=0}^p q(x) \beta_j (X_i - x_0)^j \right]^2 \\ & \left. + \left[ y_b - \sum_{j=0}^p \beta_j (X_n - x_0)^j \right]^2 \right\} K_h(X_i - x). \end{aligned} \quad (11)$$

We can write the solution vector:

$$\tilde{\beta} = (\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W \tilde{Y}, \quad (12)$$

where

$$\tilde{Y} = \begin{pmatrix} y_a \\ f(X_2) \\ f(X_3) \\ \vdots \\ f(X_{n-1}) \\ y_b \end{pmatrix}. \quad (13)$$

In order to find the expression of  $\tilde{X}$ , we need to find the elements of  $\tilde{X}$  at first.

Consider the following:

$$\begin{aligned} i = 1, \quad a_{1j} &= (X_1 - x_0)^j, \quad j = 0, \dots, p \\ i = 2, \dots, n-1, \quad b_{ij} &= j(j-1)(X_i - x_0)^{j-2}, \\ & \quad j = 2, \dots, p; \\ c_{ij} &= p(X_i) j (X_i - x_0)^{j-1}, \\ & \quad j = 0, \dots, p; \\ t_{ij} &= q(X_i) (X_i - x_0)^j, \\ & \quad j = 0, \dots, p; \\ i = n, \quad a_{nj} &= (X_n - x_0)^j, \quad j = 1, \dots, p. \end{aligned} \quad (14)$$

TABLE I: Comparison of LPR and OPA.

Parameter ( $n, p, h$ )	Residuals $e_n^2$	Pe of LPR	Pe of OPA	$h'$
$n = 21, p = 5, h = \pi/30$	$1.0435 \times 10^{-6}$	0.032	0.342	$\pi/2/100$
$n = 45, p = 6, h = \pi/60$	$4.0214 \times 10^{-8}$	0.0004	0.035	$\pi/2/1000$
$n = 60, p = 8, h = \pi/80$	$3.6503 \times 10^{-9}$	0.00001	0.003	$\pi/2/10000$

Consequently, we can get the  $\tilde{X}$

$$\tilde{X} = \begin{bmatrix} a_{10} & a_{11} & \cdots & \cdots & a_{1p} \\ -c_{20} - t_{20} & \cdots & b_{22} - c_{22} - t_{22} & \cdots & b_{2p} - c_{2p} - t_{2p} \\ -c_{30} - t_{30} & \cdots & b_{32} - c_{32} - t_{32} & \cdots & b_{3p} - c_{3p} - t_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -c_{n-1,0} - t_{n-1,0} & \cdots & b_{n-1,2} - c_{n-1,2} - t_{n-1,2} & \cdots & b_{n-1,p} - c_{n-1,p} - t_{n-1,p} \\ a_{n,0} & a_{n,1} & \cdots & \cdots & a_{n,p} \end{bmatrix}_{n,p+1}. \quad (15)$$

Substituting  $\tilde{X}$  and  $\tilde{Y}$  into

$$\tilde{\beta} = (\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T W \tilde{Y}. \quad (16)$$

Now, we can obtain the value of  $\tilde{\beta}$ .

#### 4. Example Simulation

In this part, we verified this method by solving three examples, solved the solution of specific differential problems and integration with local polynomial estimator, calculated the residual sum of squares, and compared with the existing algorithms. Calculations were performed by using MATLAB 7.0 programming solution.

##### 4.1. Initial Value Problem of Homogeneous

###### Linear Differential Equation

*Example 1.* Boundary value problem of linear differential equation with constant coefficients:

$$\begin{aligned} y''(x) + y(x) &= 0, & 0 \leq x \leq \frac{\pi}{2}, \\ y(0) &= 0, & y\left(\frac{\pi}{2}\right) = 1, \end{aligned} \quad (17)$$

The eigenvalue method can be used to obtain the real solution of initial value problem. Different parameters  $n$ ,  $h$ , and  $p$  correspond to different residual sum of square, given in Figure 1 and Table 1. LPR represents local polynomial regression. The exact solution is  $y = \sin(x)$ . In Figure 1, LPR is carried out for  $n = 21$  and  $p = 5$  giving  $h = \pi/30$ , and we get the  $e_n^2$  and percentage error (Figure 1: Solution of local polynomial estimator and exact solution of Example 1).

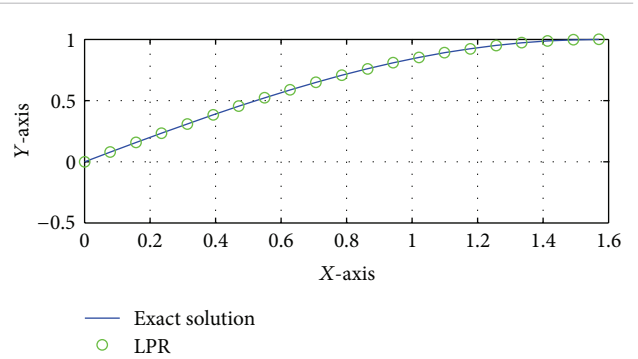


FIGURE 1: LPR of Example 1 with  $n = 21$ ,  $p = 5$  and  $h = \pi/30$ .

In Table 1, comparison of the two methods, when  $h'$  is  $\pi/2/100$ , we can get the percentage error of optimization algorithm. By comparing the two methods, we can know the difference of percentage error. Certainly, local polynomial regression has more advantages. Then, we can get better results for  $n = 45$  and  $p = 6$  given  $h = \pi/60$ ,  $n = 60$ ,  $p = 8$ , and  $h = \pi/80$ .

In Table 1, LPR represents local polynomial regression, OPA stands for optimization algorithm, Pe stands for Percentage error and  $h$  represents step size in [3]. Residuals  $e_n^2 = \sum_{i=1}^n [y(X_i) - \hat{y}(X_i)]^2$ .

*Example 2.* Boundary value problem of homogeneous linear differential equations:

$$\begin{aligned} y'' - (2 \sin x) y' - (\cos x - \sin x^2) y &= 0, & 0 \leq x \leq 1, \\ y(0) &= 0, & y(1) = e^{-\cos 1}. \end{aligned} \quad (18)$$

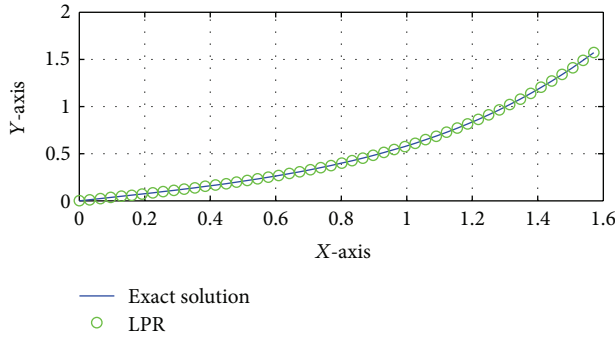


FIGURE 2: LPR of Example 2 with  $n = 50$ ,  $p = 7$  and  $h = 1/20$ .

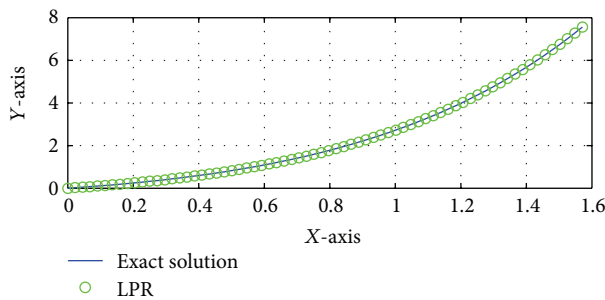


FIGURE 3: Absolute errors figure of Example 3 with  $n = 21$ ,  $p = 5$  and  $h = \pi/30$ .

TABLE 2: Local polynomial regression.

Parameter $(n, p, h)$	Residual sum of square $e_n^2$
$n = 24, p = 5, h = 1/12$	$6.2523 \times 10^{-6}$
$n = 50, p = 7, h = 1/20$	$1.5162 \times 10^{-7}$
$n = 70, p = 9, h = 1/32$	$4.4037 \times 10^{-8}$

Different parameters  $n$ ,  $h$ , and  $p$  correspond to different residual sum of square, given in Figure 2 and Table 2. LPR represents local polynomial regression. The exact solution is  $y = xe^{-\cos x}$ . In Figure 2, local polynomial estimator is carried out for  $n = 50$  and  $p = 7$  given  $h = 1/20$ , and we can get the  $e_n^2$ . Here,  $e_n^2 = \sum_{i=1}^n [y(X_i) - \hat{y}(X_i)]^2$ . We conclude that local polynomial regression is more effective. Certainly, we can get better results for  $n = 60$  and  $p = 9$  given  $h = 1/135$  (Figure 2: Solution of local polynomial estimator and exact solution of Example 2).

*Example 3.* Boundary value problem of nonhomogeneous linear differential equations boundary value problem:

$$\begin{aligned}
 xy'' + 2(1-x)y' + (x-2)y &= 2e^x, \\
 y\left(\frac{1}{2}\right) &= \frac{1}{2}e^{1/2}, \quad y(1) = e.
 \end{aligned}
 \tag{19}$$

TABLE 3: Local polynomial regression.

Parameter $(n, p, h)$	Residual sum of square $e_n^2$
$n = 20, p = 5, h = 1/11$	$5.4816 \times 10^{-6}$
$n = 40, p = 5, h = 1/19$	$3.6711 \times 10^{-7}$
$n = 70, p = 5, h = 2/67$	$8.5220 \times 10^{-8}$

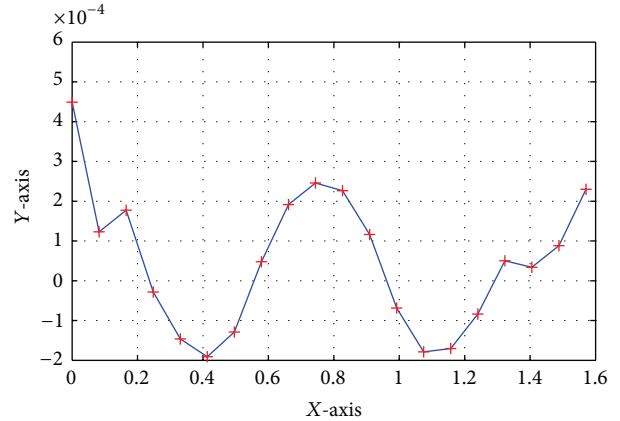


FIGURE 4: Absolute errors figure of Example 3 with  $n = 20$ ,  $p = 4$  and  $h = 1/11$ .

The exact solution is  $y = xe^x$ . Different parameters  $n$ ,  $h$ , and  $p$  correspond to different residual sum of square, given in Figure 3 and Table 3. LPR represents local polynomial regression. In Figure 3, local polynomial estimator is carried out for  $n = 20$  and  $p = 4$  giving  $h = 1/11$ , and we can get the residual sum of square  $e_n^2$  which is very small. Here,  $e_n^2 = \sum_{i=1}^n [y(X_i) - \hat{y}(X_i)]^2$ . In Table 3, local polynomial regression, it can be concluded that the effect of local polynomial regression is good and the residual sum of square  $e_n^2$  is very small. It is sure that we can get better results for  $n = 70$  and  $p = 10$  giving  $h = 0.0045$  (Figure 3: Solution of local polynomial estimator and exact solution of Example 3).

Figures 4, 5, and 6 present the absolute errors of this example under three groups of parameters. We can see that the absolute errors obtain  $\times 10^{-4}$  magnitude given different parameters  $n, h$  with the same  $p = 5$ . Moreover, with the increase in  $n$ , the absolute errors decrease gradually. From Examples 1–3, we can conclude that the residual sum of square and absolute error are smaller by using local polynomial method which shows good accuracy.

## 5. Conclusions

In this paper, a local polynomial-based fitting method has been proposed for numerical solutions of differential equations with boundary values. Comparisons of the computed results with exact solutions and existing methods showed that the method has the capability of solving the differential equations with boundary values. It is also capable of obtaining

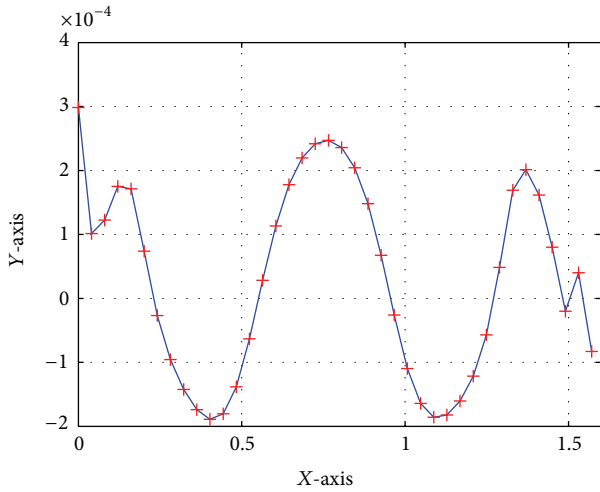


FIGURE 5: Absolute errors figure of Example 3 with  $n = 40$ ,  $p = 6$  and  $h = 1/19$ .

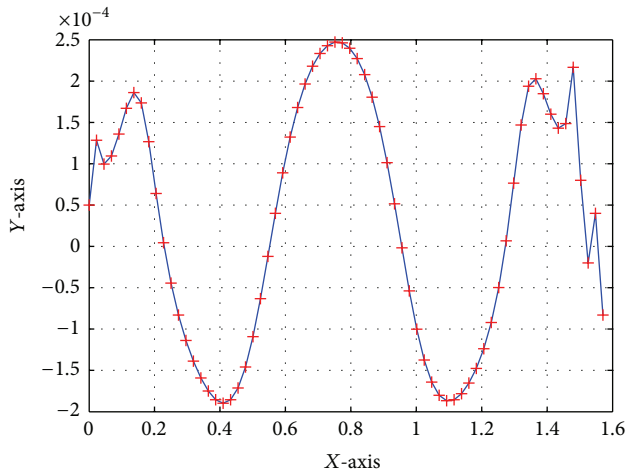


FIGURE 6: Absolute errors figure of Example 3 with  $n = 70$ ,  $p = 10$  and  $h = 2/67$ .

highly accurate results with minimal computational effort for both time and space. It was seen that the local polynomial-based technique approximates the exact solution very well. Numerical results demonstrated that the proposed method is more effective, and local polynomial estimator is more accurate compared with other algorithms, for instance, optimization algorithm. Furthermore, by using these methods, we can also study the numerical solutions of boundary value problems for differential equations on whole line or fractional differential equations which have been studied by the authors in [11–13]. In order to illustrate our method further, we will focus on these kinds of fractional differential, integral differential equations.

### Acknowledgments

This work was supported by the Natural Science Foundation Projects of CQ CSTC of China (CSTC2010BB2310,

CSTC2011jjA40033, and CSTC2012jjA00037) and Chongqing CMEC Foundation of China (KJ120829).

### References

- [1] C. P. Gupta, “Solvability of a three-point nonlinear boundary value problem for a second order ordinary differential equation,” *Journal of Mathematical Analysis and Applications*, vol. 168, no. 2, pp. 540–551, 1992.
- [2] C. P. Gupta, “A generalized multi-point boundary value problem for second order ordinary differential equations,” *Applied Mathematics and Computation*, vol. 89, no. 1–3, pp. 133–146, 1998.
- [3] W. WenXia, “A class of nonlinear integral boundary value problems of differential equations,” *Journal of Engineering Mathematics*, vol. 22, no. 3, pp. 555–558, 2005.
- [4] X. L. Hou, Y. Qian, and H. T. Wu, “Optimization algorithm for boundary value problems for nonlinear ordinary differential equations,” *Chinese Journal of Engineering Mathematics*, vol. 27, no. 4, pp. 663–668, 2010.
- [5] 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.
- [6] Z. L. Wei, “Positive solutions of some singular sublinear three-point boundary value problems,” *Acta Mathematica Scientia A*, vol. 28, no. 1, pp. 174–182, 2008.
- [7] 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.
- [8] L.-Y. Su, “Prediction of multivariate chaotic time series with local polynomial fitting,” *Computers and Mathematics with Applications*, vol. 59, no. 2, pp. 737–744, 2010.
- [9] L. Su, “Multivariate local polynomial regression with application to Shenzhen component index,” *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 930958, 11 pages, 2011.
- [10] J. Fan and Q. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*, Springer, 2003.
- [11] L. Yuji, “Existence and uniqueness of solutions for initial value problems of multi-order fractional differential equations on the half lines,” *Scientia Sinica A*, vol. 42, no. 7, pp. 735–756, 2012 (Chinese).
- [12] Y. Liu, “Multiple bounded positive solutions to integral type BVPs for singular second order ODEs on the whole line,” *Abstract and Applied Analysis*, vol. 2012, Article ID 352159, 23 pages, 2012.
- [13] Y. Liu, “Non-homogeneous boundary-value problems of higher order differential equations with  $p$ -Laplacian,” *Electronic Journal of Differential Equations*, vol. 2008, no. 20, pp. 1–43, 2008.



# Hindawi

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

