Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 565896, 12 pages
doi:10.1155/2012/565896

*Research Article*

# Error Upper Bounds for a Computational Method for Nonlinear Boundary and Initial-Value Problems

**Osvaldo Guimarães and José Roberto C. Piqueira**

*Escola Politécnica da Universidade de São Paulo, Avenida Professor Luciano Gualberto, Travessa 3, No. 158, 05508-900 São Paulo, SP, Brazil*

Correspondence should be addressed to José Roberto C. Piqueira, piqueira@lac.usp.br

Received 7 December 2011; Revised 16 January 2012; Accepted 17 January 2012

Academic Editor: Jyh Horng Chou



This work develops a computational approach for boundary and initial-value problems by using operational matrices, in order to run an evolutive process in a Hilbert space. Besides, upper bounds for errors in the solutions and in their derivatives can be estimated providing accuracy measures.

## 1. Introduction

Differential equations are ubiquitous in engineering daily life but their solutions are sometimes very difficult, mainly if they are nonlinear. Several of them do not have an analytical solution describable by a finite combination of elementary functions, or even by an unlimited series with a determinable recurrence relation.

In previous works, analytical and numerical results were obtained for nonlinear differential equations [1–3], and an algorithm called SIV (Solving Initial Value) was developed. Here, the problem of determining the error limits for SIV is emphasized, providing quality parameters for the method.

In Section 2, some general theoretical considerations about a numerical method for differential equations, by using expansions in Hilbert space, are presented. Error upper bounds estimation is discussed in Section 3, including some examples. Section 4 concludes the reasoning.

## 2. Methodology

Problems described by differential equations can be submitted to initial conditions and, in those cases, they are called initial-value problems (IVP) and there are a number of softwares

dedicated to their solutions, based on Runge-Kutta and Adams methods [4] constantly improved [5, 6], mainly considering applications to physics problems [7–10].

Lipschitz theorem gives a sufficient condition for a differential equation to have a unique solution and, under the theorem assumptions, mass matrix is guaranteed to be nonsingular, providing that the classical methods can start running and obtaining, for sequential intervals, continuous expansions for limited functions [11]. The precision is only limited by the computational apparatus.

Besides IVP problems, there are the boundary value problems BVP that, mainly, have been solved by using numerical methods derived from Galerkin and variational methods [12, 13]. Previous works [1–3] developed analytical and numerical methods, based on a spectral approach, that can be applied either to IVP or to BVP.

Details of the method, called SIV, are described mainly in [3]. Here, some hints are presented for the sake of clarity, in order to discuss precision issues.

The implemented evolutive algorithm is genetic and differential [14], conceived for an evolution on a Hilbert space, with which gene is represented by the coefficients of a possible expansion in terms of a basis of this space and corresponding to a sixty-individual vector population (chromosomes). Each generation is submitted to ranking contests.

The error obtained substituting the vector in the equation is the criterium to determine the individual ranking position. The lesser the error is, the better is the individual ranking position. Selecting the reproducers for the next generation is performed by attributing greater choice probability to the best ranked individuals [3].

Heritage is simulated by taking two vectors and combining them to generate a descendent that is a new vector with the same dimension of the original vectors. The criteria to choose if the gene in a locus is given by one or another original vector are the cross-over with the loci randomly chosen [15].

The introduction of learning rules in the algorithm [1, 3] reduced about a hundred times the processing time because the search space is continually contracted with error margins decreasing.

Considering the solution $f(u)$ to be continuous and expressed as a linear combination of the elements from an orthogonal basis $B_k(u)$ in the Hilbert functional space is the main assumption of the method presented in [3].

Under these conditions, $f(u) = \sum_{k=0}^{n} c_k B_k(u)$, with $c_k = \langle f \mid B_k \rangle$, $\langle B_j \mid B_k \rangle = g(k)\delta_{i,j}$, and the symbol $\langle \cdot \mid \cdot \rangle$ represents the internal product in the Hilbert space [3].

Defining $\langle f \mid B_k \rangle = \int_a^b f \cdot B_k w(u) du$ with $w(u)$ being the nonnegative weight function, considering the interval $[-1, 1]$ and $w(u) = 1$, the Legendre polynomial basis is obtained with $g(k) = 2/(2k + 1)$. For the same interval and by using generic $w(u)$, Jacobi polynomials are obtained. Legendre and Chebyshev polynomials are particular cases of Jacobi polynomials. For half-limited intervals, the Laguerre polynomials are used; for unlimited intervals, the Hermite polynomials are adequate.

Solving differential equations with orthogonal series approximations have been thoroughly studied in the last three decades, providing efficient algorithms [16–19]. In this approach, it is possible to transform a differential equation into an algebraic one, giving the coefficients of the series expansion. This transformation is performed by using operational matrices, extensively discussed in [3].

Completeness and orthogonality in Hilbert spaces are related to certain domains, depending on the basis chosen. Consequently, it is important to develop basis transformations to consider this fact. In [1, 3] a method for transposing domains changing a variable that transforms the differential equation was developed. Besides, initial and boundary conditions

need to be transposed in the same way and the program described in [1, 3] takes care of this transposition, too.

## 3. Error Upper Bounds

One of the main problems in the numerical solutions of differential equations is how to estimate the error margins. In the majority of the cases, it is possible to stipulate the maximum tolerable error for each step but, as the integration for the whole interval comprehends thousands of steps, the cumulative error is difficult to be obtained [20].

Here, a method to determine the upper bounds for the absolute error in an integration process is described, even if the algorithm and the analytical solution are unknown. These ideas can be used, in order to estimate the confidence for numerical approximations of the solution [21].

Starting with the differential equation written in the following form:

$$\frac{d^q y_t}{du^q} = G\left(y_t, y_t^{(1)}, y_t^{(2)}, \ldots, y_t^{(q-1)}, u\right), \tag{3.1}$$

the polynomial solution found is replaced on $G(y_t, y_t^{(1)}, y_t^{(2)}, \ldots, y_t^{(q-1)}, u)$, evaluating $G$ for the whole interval. This reasoning allows the calculation of the relative dispersion of the coefficients of the series that approximates the solution. In order to illustrate the process, Example 1 is developed.

### 3.1. Example 1

As an example of dispersion calculation, the boundary value problem given by

$$D_4 + D_3 D_1 - \frac{D_0}{2} + 3\sin^2 x + (4D_0 + 3)\cos x + (D_1 - \sin x)^2 + \frac{D_2}{2} = 0 \tag{3.2}$$

is considered, with $D_i$ representing the $i$-derivative of the function to be found.

The domain is $x \in [0, \pi]$; the boundary conditions are $y(0) = y(\pi) = y'(0) = 0$ and $y'(\pi) = -\pi$, with analytical solution being $y(x) = x \sin x$.

Transposing the domain to the interval $[-1, 1]$, by using the methodology described in [3, Sections 3.3 and 5], (3.2) becomes

$$\begin{aligned} &\sin\left(\pi\frac{(u+1)}{2}\right) - \left(\frac{2D_{1t}}{\pi}\right)^2 + \cos\left(\pi\frac{(u+1)}{2}\right)(4D_{0t} + 3) + \frac{2D_{2t}}{\pi^2} \\ &\quad + \frac{16D_{4t}}{\pi^4} + 3\sin\left(\pi\frac{(u+1)}{2}\right)^2 + \frac{16D_{1t}D_{3t}}{\pi^4} = 0, \end{aligned} \tag{3.3}$$

with boundary conditions being $y_t(-1) = y_t(+1) = y_t'(-1) = 0$, and $y_t'(+1) = -\pi^2/2$.

Then, isolating the major order derivative, choosing an order 12 Legendre series given in Table 1 from a first interaction of the SIV algorithm, described in [3, appendix B], as a solution candidate, and applying the same SIV procedure [3] to integrate it four times, the function shown in Figure 1 is obtained.

**Table 1:** Order 12 Legendre series for Example 1.

| Order | Coefficient |
|---|---|
| 0 | $1.0000E+00$ |
| 1 | $5.6829E-01$ |
| 2 | $-1.0793E+00$ |
| 3 | $-6.1141E-01$ |
| 4 | $8.1334E-02$ |
| 5 | $4.4221E-02$ |
| 6 | $-2.0899E-03$ |
| 7 | $-1.1126E-03$ |
| 8 | $2.6910E-05$ |
| 9 | $1.4139E-05$ |
| 10 | $-2.0746E-07$ |
| 11 | $-1.0491E-07$ |
| 12 | $1.0046E-09$ |

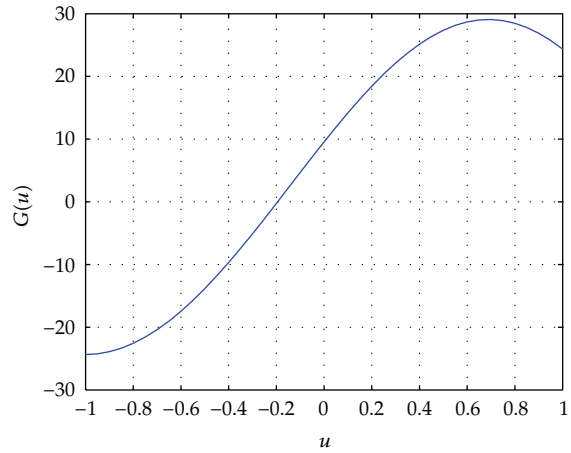



**Figure 1:** $G(u)$ for Example 1.

After the integrations, the coefficients of the result are compared with the coefficients of the candidate and the dispersion for each coefficient is determined. In spite of not having a statistical meaning, the term dispersion is used, for the sake of simplicity.

It is worth noticing that, for the Legendre basis $\int_{-1}^{1}[P_k(u)]^2du = 2/(2k+1)$, and, consequently, increasing the order, this term decreases meaning that higher-order coefficients with the same absolute errors present greater relative errors.

Consequently, the dispersion to be exhibited follows the following relation:

$$\sigma_k^{\text{rel}} = \sigma_k\sqrt{\frac{2k+1}{2}}. \tag{3.4}$$

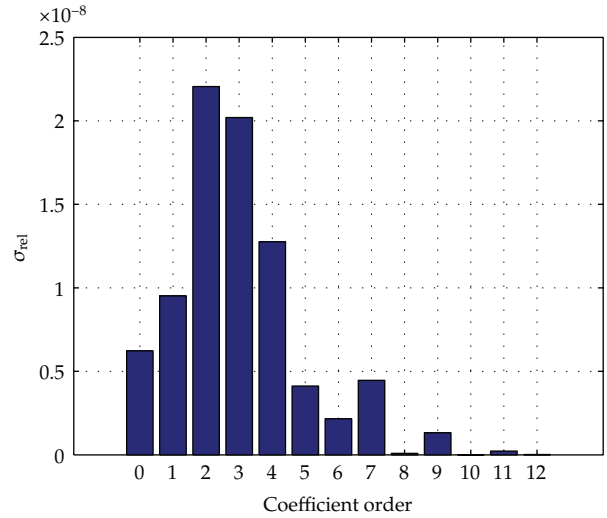



**Figure 2:** Relative dispersion of the coefficients (Example 1).

Figure 2 shows the relative dispersion for the coefficients of the approximated solution presented in Example 1. This dispersion calculation is used to estimate the error upper bounds, as shown in the following.

## 3.2. Calculating Error Upper Bounds

### 3.2.1. Solution Error Upper Bound

Considering $f(u)$ given by a Legendre series approximation $f^*(u)$, up to order $n$, one can write $f(u) = \sum_{k=0}^{\infty} c_k P_k$ and $f^*(u) = \sum_{k=0}^{n} (c_k + \delta_k) P_k$, with $\delta k$ being the error affecting $c_k$.

Consequently, the total error up to order $n$ is given by

$$\delta f(u) = f^*(u) - f(u) = \sum_{k=0}^{n} (c_k + \delta_k - c_k) P_k = \sum_{k=0}^{n} \delta_k P_k \Longrightarrow \delta f(u) \leq \sum_{k=0}^{n} |\delta_k P_k| \leq \sum_{k=0}^{n} |\delta_k P_{k,\max}|. \tag{3.5}$$

Considering the worst scenario, all $\delta_k$ will contribute to increase the error exactly where $P_k$ is maximum. As a matter of fact, some increase the error and some decrease it. In any case, the error upper bound is given by (3.5) and the maximum value of $P_k$, for any $k$, is 1.

Consequently, the maximum error in the polynomial approximation is

$$|\delta|_{\max} = \sum_{k=0}^{n} |\delta_k|. \tag{3.6}$$

### 3.2.2. Derivative Error Upper Bounds

For each derivative, the error bound calculation is analogous to the procedure for the solution error. For the $j$-order derivatives, the error can be written as $\delta f^j(u) = f^{*j}(u) - f^j(u)$, for $j$ from 1 to the differential equation order.

This calculation can be made by using the differentiation matrix $M_{LD}$ [2]. Denoting the exponent of the differentiation matrix by the number of times that was applied to the coefficients vector,

$$\delta f_k^j(u) = \left[M_{LD}^j(c_k)^T - M_{LD}^j(c_k + \delta_k)^T\right] P_k, \tag{3.7}$$

for $j$ from 1 to the differential equation order. Considering $P_{k,\max} = 1$,

$$\delta f_k^j(u) = [1 \cdots 1] M_{LD}^j(\delta_k)^T. \tag{3.8}$$

The error upper bounds in the derivatives are slightly greater than those in the solutions, because the derivatives are expressed by lower order polynomials. In spite of this, the error margins are very low. As a matter of illustration, if the solution of order 7 is used, the error upper bounds for the first, second, and third derivatives are,

$$\begin{aligned} \delta f_{\max}^1 &= \delta_1 + 3\delta_2 + 6\delta_3 + 10\delta_4 + 15\delta_5 + 21\delta_6 + 28\delta_7, \\ \delta f_{\max}^2 &= 3\delta_2 + 15\delta_3 + 45\delta_4 + 105\delta_5 + 210\delta_6 + 378\delta_7, \\ \delta f_{\max}^3 &= 15\delta_3 + 105\delta_4 + 420\delta_5 + 1260\delta_6 + 3150\delta_7, \end{aligned} \tag{3.9}$$

with $\delta_i$ representing the coefficient errors.

### 3.2.3. Transposing Error Upper Bounds

The expressions for the error upper bounds developed in the former sections are for the work domain, but it is important to transpose them for the domain of the original equation. The domain transposition method developed in [1, 3] can be applied to the errors.

For instance, in order to transpose the first-order derivative error, it can be written as $D_1 = D_{1t}\lambda$, with $du/dx = \lambda$ and, therefore

$$\delta_1 = \delta_{1t}\lambda. \tag{3.10}$$

For the second-order derivative,

$$D_2 = D_{2t}\lambda^2 + D_{1t}\frac{d\lambda}{du}\lambda \Longrightarrow \delta_2 = \delta_{2t}\lambda^2 + \delta_{1t}\frac{d\lambda}{du}\lambda. \tag{3.11}$$

For superior-order derivatives, the reasoning is analogous.

**Table 2:** Error upper bounds for Example 1.

| Function | Error upper bound |
|---|---|
| $D_0$ | $5.21E-08$ |
| $D_1$ | $1.84E-07$ |
| $D_2$ | $8.32E-07$ |
| $D_3$ | $4.44E-06$ |
| $D_4$ | $2.38E-05$ |

### 3.3. Finishing Example 1

For Example 1, the upper bounds of the errors for the Legendre series in the interval $[-1,1]$ can now be calculated by using the transposing procedures described previously and are shown in Table 2. Additionally, Figure 3 shows the local errors along the interval. Again, the SIV procedure described in [3] is used, in order to search the solutions.

### 3.4. Example 2

In the following example, the tools described here and in [1–3] will be applied to a boundary value problem, running in 4 GB RAM processor, and taking 20 s for the whole processing. The differential equation to be solved is

$$D_4 + \frac{4D_1\left(3D_1x\left(x^4-1\right)-9x^2+2\right)}{\left(1+x^2\right)^3\arctan(x)} = 0, \tag{3.12}$$

with $x \in [0,1]$ and $y(0) = y = 0$; $y(1) = \pi^2/16$ and $y(1) = \pi/4$, with $D_i$ representing the $i$-derivative of the function to be found.

Under these conditions, the analytical solution is $y = \arctan^2(x)$, and, after 100 generations of the SIV algorithm described in [3, appendix B], in the Jacobi polynomial domain, the dispersion of the coefficients is shown in Figure 4. The residual values when the approximated solution is replaced in (3.12) are shown in Figure 5.

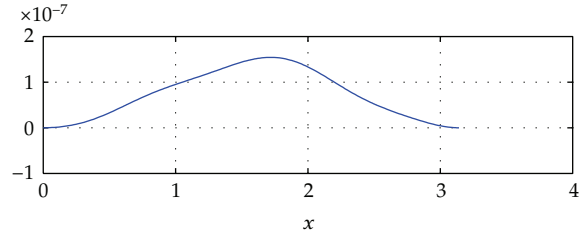
The error upper bounds are given in Table 3, the obtained solution is

$$f^*(x) = \arctan^2(x) + 3.5x10^{-17}\arctan(x) + 5.7x10^{-16}\arctan^3(x) + 2.8x10^{-18}, \tag{3.13}$$

and considering that the analytical solution is given, the local error for the interval can be found and is shown in Figure 6.

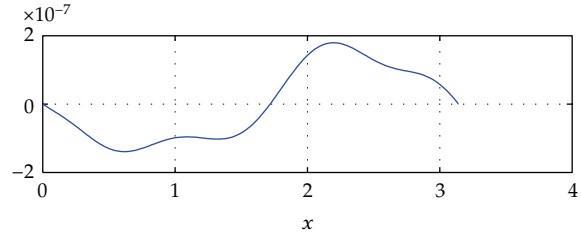
## 4. Conclusions

Since the seminal work of Chen and Hsiao [22], the research about operational matrices and the computational capacity advances permitted a strong development in numerical solutions for differential equations. The SIV algorithm developed in [1, 3] gives an interesting combination of operational matrices and computational techniques, by using polynomial approximations for the solutions.



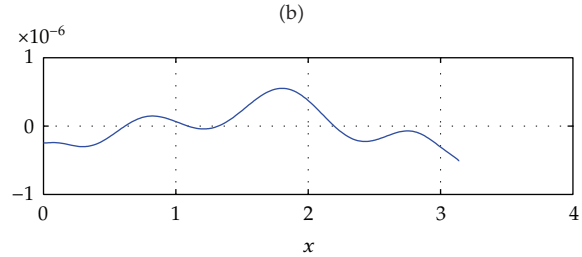

Error in $D_0$

(a)



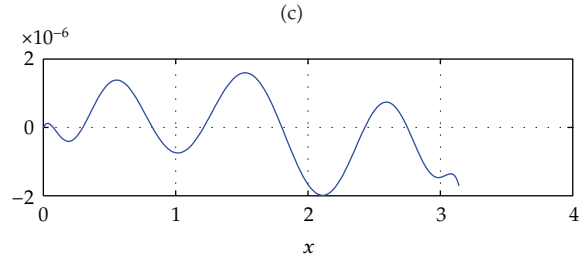

Error in $D_1$

(b)



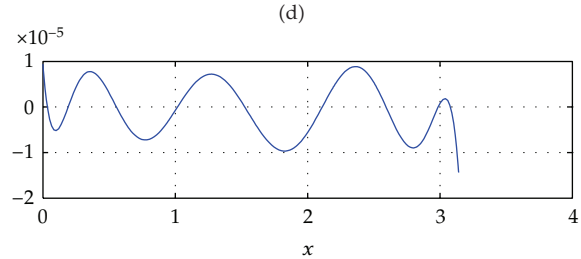

Error in $D_2$

(c)




Error in $D_3$

(d)




Error in $D_4$

(e)

**Figure 3:** Local errors (Example 1).

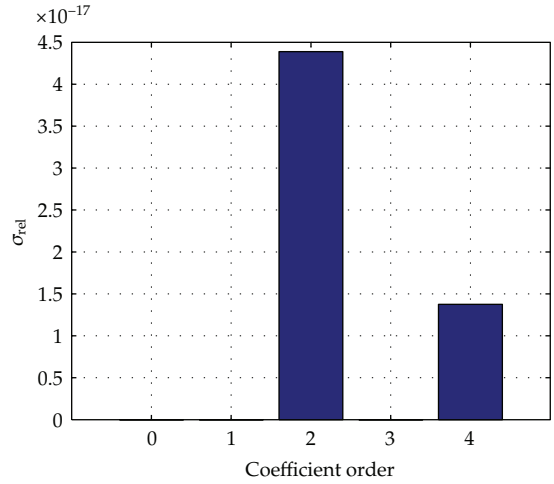



**Figure 4:** Relative dispersion of the coefficients (Example 2).

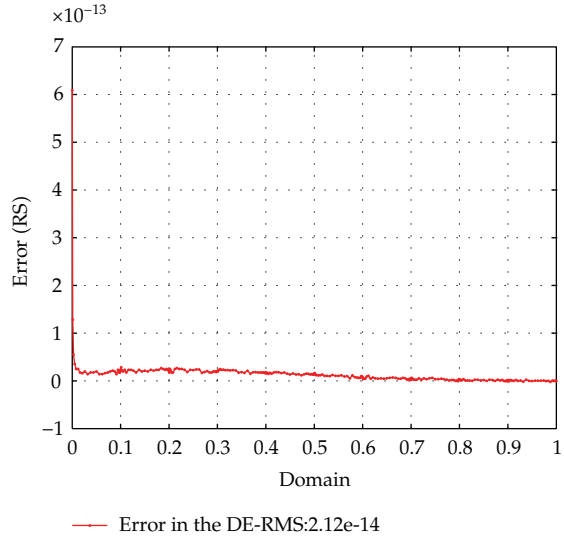



**Figure 5:** Residual values (Example 2).

**Table 3:** Error upper bounds for Example 2.

| Function | Error upper bound |
|---|---|
| $D_0$ | $9.56E-018$ |
| $D_1$ | $1.54E-016$ |
| $D_2$ | $9.92E-016$ |
| $D_3$ | $4.23E-015$ |
| $D_4$ | $1.98E-013$ |

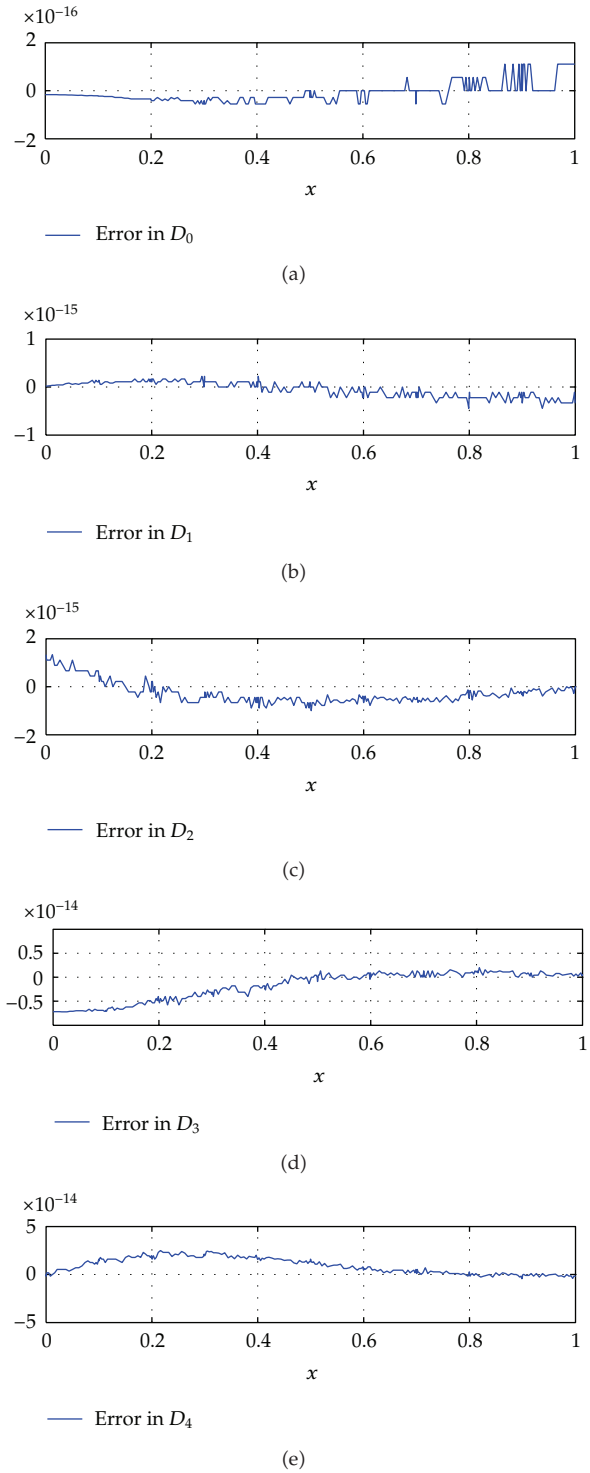



**Figure 6:** Local errors (Example 2).

Several advantages are inherent to the method:

(i) the basis of the Hilbert spaces is complete and the solutions are square integrable; consequently, increasing the order of the series, the approximation is improved;

(ii) as the elements of the basis are orthogonal, changing the coefficients of a component does not interfere with the precision of the other components;

(iii) the learning process of the algorithm reduces the search space, reducing the computational costs and decreasing the error margins;

(iv) the error bounds can be determined in a very simple way.

## Acknowledgment

This work is supported by CNPq.

## References

[1] O. Guimarães, *Computação evolutiva na resolução de equações diferenciais ordinárias não lineares no espaço de Hilbert*, Ph.D. thesis, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil, 2008.
[2] O. Guimarães, J. R. C. Piqueira, and M. Lobo-Netto, "Direct computation of operational matrices for polynomial bases," *Mathematical Problems in Engineering*, vol. 2010, Article ID 139198, 12 pages, 2010.
[3] O. Guimarães, J. R. C. Piqueira, and M. Lobo-Netto, "Combining Legendre's polynomials and genetic algorithm in the solution of nonlinear initial-value problems," *Mathematical Problems in Engineering*, vol. 2011, Article ID 521342, 20 pages, 2011.
[4] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
[5] R. H. Battin, "Resolution of Runge-Kutta-Nystrom condition equations through eighth order," *American Institute of Aeronautics and Astronautics Journal*, vol. 14, no. 8, pp. 1012–1021, 1976.
[6] R. H. Battin, "Resolution of Runge-Kutta-Nystrom condition equations through 8th order—reply," *American Institute of Aeronautics and Astronautics Journal*, vol. 15, no. 5, pp. 763–763, 1977.
[7] G. Psihoyios and T. E. Simos, "A fourth algebraic order trigonometrically fitted predictor-corrector scheme for IVPs with oscillating solutions," *Journal of Computational and Applied Mathematics*, vol. 175, no. 1, pp. 137–147, 2005.
[8] Z. A. Anastassi and T. E. Simos, "An optimized Runge-Kutta method for the solution of orbital problems," *Journal of Computational and Applied Mathematics*, vol. 175, no. 1, pp. 1–9, 2005.
[9] T. E. Simos, "Closed Newton-Cotes trigonometrically-fitted formulae of high order for long-time integration of orbital problems," *Applied Mathematics Letters*, vol. 22, no. 10, pp. 1616–1621, 2009.
[10] T. E. Simos, "Exponentially and trigonometrically fitted methods for the solution of the Schrödinger equation," *Acta Applicandae Mathematicae*, vol. 110, no. 3, pp. 1331–1352, 2010.
[11] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997.
[12] C. L. Karr, I. Yakushin, and K. Nicolosi, "Solving inverse initial-value, boundary-value problems via genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 6, pp. 625–633, 2000.
[13] L. P. Lara and M. Gadella, "An approximation to solutions of linear ODE by cubic interpolation," *Computers & Mathematics with Applications*, vol. 56, no. 6, pp. 1488–1495, 2008.
[14] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
[15] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1997.
[16] M. Razzaghi and S. Yousefi, "The Legendre wavelets operational matrix of integration," *International Journal of Systems Science*, vol. 32, no. 4, pp. 495–502, 2001.
[17] E. M. E. Elbarbary, "Legendre expansion method for the solution of the second- and fourth-order elliptic equations," *Mathematics and Computers in Simulation*, vol. 59, no. 5, pp. 389–399, 2002.

[18] E. H. Doha and A. H. Bhrawy, "Efficient spectral-Galerkin algorithms for direct solution of fourth-order differential equations using Jacobi polynomials," *Applied Numerical Mathematics*, vol. 58, no. 8, pp. 1224–1244, 2008.
[19] F. Khellat and S. A. Yousefi, "The linear Legendre mother wavelets operational matrix of integration and its application," *Journal of the Franklin Institute—Engineering and Applied Mathematics*, vol. 343, no. 2, pp. 181–190, 2006.
[20] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists*, Academic Press, San Diego, Calif, USA, 4th edition, 1995.
[21] M. G. Armentano, "Error estimates in Sobolev spaces for moving least square approximations," *SIAM Journal on Numerical Analysis*, vol. 39, no. 1, pp. 38–51, 2001.
[22] C. F. Chen and C. H. Hsiao, "Walsh series analysis in optimal control," *International Journal of Control*, vol. 21, no. 6, pp. 881–897, 1975.