TAYLOR-SERIES EXPANSION METHOD FOR VOLTERRA INTEGRAL EQUATIONS OF THE SECOND KIND

P. HUABSOMBOON[1] B. NOVAPRATEEP[1,2] AND HIDEAKI KANEKO[3,*]

Received August 30, 2010

ABSTRACT. In this paper, we continue our study that began in recent papers [2] and [3] concerning a simple yet effective Taylor series expansion method to approximate a solution of integral equations. The method is applied to Volterra integral equation of the second kind as well as to systems of Volterra equations. The results obtained in this paper improve significantly the results reported in recent papers, [7] and [8]. An error analysis is also provided.

1 Introduction In a recent paper [2], a Taylor-series expansion method for approximating the solutions of the second kind Fredholm integral equations was established. Subsequently, the method was extended in [6] to approximating the solutions of a system of second kind Fredholm integral equations. A success of the Taylor-series expansion method developed in [2] relies heavily upon the property that a kernel $k(|t-s|)$ of convolution type decays rapidly as $|t-s|$ increases. Their method, therefore, not only does not apply to a wider class of the second kind Fredholm integral equations having kernels of different variations but also the accuracy of approximation depends upon the rate at which a convolution kernel approaches zero as $|t-s| \to \infty$. The present authors [3] recently generalized the results in [2] and [6] to obtain a new Taylor-series expansion method which is applicable to a larger class of Fredholm equations as well as is capable of delivering more accurate approximations. Also, in [4], the results in [3] were extended to obtain an approximation of the solution of nonlinear Hammerstein equation. One of the important points made in [3] and in [4] is that the the Taylor-series expansion method lends naturally into parallel computation environment. This is due largely to the fact that approximations by the Taylor-series expansion method of the solution of the second kind Fredholm equations can be made pointwise. Note that most numerical methods for approximating the solutions of integral equations are global in nature, -i.e., approximation must be made over the entire interval. This results in a system of linear or nonlinear equations which are normally of large scale and dense, and therefore expensive to solve, for references see, -e.g., [1] for linear equations and [5] and references cited within for nonlinear equations. Thus, the characteristics that the Taylor-series expansion method can approximate the solution pointwise is an important point since, besides the parallel computation environment it induces, there creates a possibility of finding the solution of integral equations only over a small region of interest. It is also important to note that the methods explored in [3], [4] and in this paper compute concurrently accurate approximations of the derivatives of the solution as well.

The purpose of this paper is to apply the method established in [3] to obtaining a solution of the second kind Volterra integral equation and solution of a system of the second kind Volterra integral equations. Our numerical experiments performed on the examples in [7] and [8] demonstrate that our results are significantly better than the results reported in these papers. This paper is organized as follows: In Section 2, we describe our Taylor-series expansion method for the second kind Volterra integral equations. An error analysis is also given in this paper. The method is then extended, in Section 3, to solve a system of the second kind Volterra integral equations. Numerical examples are given at the end of each section.

2000 *Mathematics Subject Classification.* 65R20,45D05 .
Key words and phrases. Taylor-expansion method, Volterra integral equations.

2 Taylor-series Expansion Method We consider the following Volterra integral equation of the second kind,

$$x(s) - \lambda \int_0^s k(s,t)x(t)dt = y(s), \qquad 0 \le s \le T, \tag{2.1}$$

where $T > 0$, $y \in C^n[0,T]$ and $k \in C^n([0,T] \times [0,T])$, $n \ge 1$ are known functions, the parameter λ is given and x is a function to be determined. We do not consider the case for Volterra equation with weakly singular kernel in this paper. We note that even though weakly singular equations are discussed in [2], [6] and [7], the numerical results reported in these papers are not very good in these settings. This is due to the fact that the current method as well as the methods discussed in [2], [6] and [7] applies to weakly singular equations only at the lowest order accuracy.

Let $k_s^{(i)}(s,t) = \frac{\partial^i k(s,t)}{\partial s^i}$, $i = 1, \dots, n$ and $k_s^{(i)}(s,s) = \frac{\partial^i k(s,t)}{\partial s^i}|_{t=s}$. Also, let $[k_s^{(i)}(s,s)]^{(j)} = \frac{d^j}{ds^j} k_s^{(i)}(s,s)$ and $k^{(j)}(s,s) = \frac{d^j}{ds^j} k(s,s)$, $j = 0, 1, \dots, n$. Differentiating (2.1) n times, we obtain

$$\begin{aligned}
x'(s) - \lambda\{k(s,s)x(s) + \textstyle\int_0^s k_s'(s,t)x(t)dt\} &= y'(s) \\
x''(s) - \lambda\{k(s,s)x'(s) + k'(s,s)x(s) + k_s'(s,s)x(s) & \\
+ \textstyle\int_0^s k_s''(s,t)x(t)dt\} &= y''(s) \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots & \\
x^{(n)}(s) - \lambda\{\sum_{j=0}^{n-1} \binom{n-1}{j} k^{(n-1-j)}(s,s)x^{(j)}(s) & \\
+ \sum_{i=0}^{n-3} \sum_{j=0}^{n-2-i} \binom{n-2-i}{j} [k_s^{(i+1)}(s,s)]^{(n-2-i-j)} x^{(j)}(s) & \\
+ \textstyle\int_0^s k_s^{(n)}(s,t)x(t)dt\} &= y^{(n)}(s).
\end{aligned} \tag{2.2}$$

We point out that in equation (5), [7], $k^{(i)}(s,s)$ and $k_s^{(i)}(s,s)$ are not distinguished. This may be one of the causes of low accurate numerical results reported in [7] even for high orders Taylor-series expansion method.

For each arbitrary but fixed s,

$$x(t) \approx x(s) + x'(s)(t-s) + \cdots + \frac{1}{n!} x^{(n)}(s)(t-s)^n. \tag{2.3}$$

Substituting (2.3) into (2.1) and into each equation in (2.2), we obtain

$$
\begin{aligned}
[1-\lambda \textstyle\int_0^s k(s,t)dt]x(s) - \lambda \sum_{i=1}^n \{\frac{1}{i!}\int_0^s k(s,t)(t-s)^i dt\}x^{(i)}(s) &\approx y(s),\\
-\lambda[k(s,s)+\textstyle\int_0^s k'_s(s,t)dt]x(s) + [1-\lambda\int_0^s k'_s(s,t)(t-s)dt]x'(s) &\\
-\lambda\sum_{i=2}^n \int_0^s k'_s(s,t)\frac{(t-s)^i}{i!}dt\,x^{(i)}(s) &\approx y'(s)\\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots &\\
-\lambda\sum_{i=0}^{n-1}\{\begin{pmatrix} n-1 \\ i\end{pmatrix}k^{(n-1-i)}(s,s) &\\
+\sum_{j=0}^{n-2-i}\begin{pmatrix} n-2-j \\ i\end{pmatrix}[k_s^{(j+1)}(s,s)]^{(n-2-i-j)} &\\
+\tfrac{1}{i!}\textstyle\int_0^s k_s^{(n)}(s,t)(t-s)^i dt\}x^{(i)}(s) &\\
+[1-\lambda\textstyle\int_0^s k_s^{(n)}(s,t)\frac{(t-s)^n}{n!}dt]x^{(n)}(s) &\approx y^{(n)}(s).
\end{aligned}
\tag{2.4}
$$

In the last equation in (2.4), we assume $\begin{pmatrix} 0 \\ 0 \end{pmatrix} \equiv 0$, $(i=0,\ j=n-2)$, and there is no term in the sum $\sum_{j=0}^{n-2-i}$ when $i=n-1$. Note that in [8] and [7], the Taylor expansion (2.3) is only used to replace $x(t)$ in equation (2.1) and each $x(t)$ in equations (2.2) is replaced by $x(s)$. This amounts to approximating x by constant function which results in lower order accuracy in numerical solutions.

For an error analysis of the current Taylor-series method, let $\bar{x}(s), \bar{x}'(s), \ldots, \bar{x}^{(n)}(s)$ be the functions that satisfy equations (2.1) and (2.4). Replacing $x(t)$ in (2.1) and (2.2) by $\sum_{i=0}^n \frac{x^{(i)}(s)}{i!}(t-s)^i + \frac{x^{(n+1)}(\xi(s))}{(n+1)!}(t-s)^{n+1}$, and subtracting them from the corresponding equations in (2.1) and (2.4), we obtain

$$
[1-\lambda\int_0^s k(s,t)dt](x(s)-\bar{x}(s)) - \lambda\sum_{i=1}^n \frac{1}{i!}\int_0^s k(s,t)(t-s)^i dt(x^{(i)}(s)-\bar{x}^{(i)}(s))
$$
$$
= \lambda\frac{x^{(n+1)}(\xi(s))}{(n+1)!}\int_0^s k(s,t)(t-s)^{(n+1)}dt
$$

$$
-\lambda[k(s,s)+\int_0^s k'_s(s,t)dt](x(s)-\bar{x}(s)) + [1-\lambda\int_0^s k'_s(s,t)(t-s)dt](x'(s)-\bar{x}'(s))
$$
$$
-\lambda\sum_{i=2}^n \frac{1}{i!}\int_0^s k'_s(s,t)(t-s)^i dt(x^{(i)}(s)-\bar{x}^{(i)}(s))
$$
$$
= \lambda\frac{x^{(n+1)}(\xi(s))}{(n+1)!}\int_0^s k'_s(s,t)(t-s)^{(n+1)}dt
$$

Similarly for the 3rd, ..., (n-1)th equations and finally for the nth equation, we obtain

$$-\lambda\sum_{i=0}^{n-1}\left\{\begin{pmatrix} n-1 \\ i \end{pmatrix} k^{(n-1-i)}(s,s)+\sum_{j=0}^{n-2-i}\begin{pmatrix} n-2-j \\ i \end{pmatrix}[k_s^{(j+1)}(s,s)]^{(n-2-i-j)}\right.$$
$$\left.+\frac{1}{i!}\int_0^s k_s^{(n)}(s,t)(t-s)^i dt\right\}(x^{(i)}(s)-\bar{x}^{(i)}(s))$$
$$+[1-\lambda\int_0^s k_s^{(n)}(s,t)\frac{(t-s)^n}{n!}dt](x^{(n)}(s)-\bar{x}^{(n)}(s))$$
$$=\lambda\frac{x^{(n+1)}(\xi(s))}{(n+1)!}\int_0^s k_s^{(n)}(s,t)(t-s)^{(n+1)}dt$$

The above equations represent a system of linear equations for the errors $x^{(i)}(s)-\bar{x}^{(i)}(s)$, for $i=0,1,\ldots n$. More specifically, with obvious assignments of a_{ij} from the above equations, -e.g., $a_{00}=1-\lambda\int_0^s k(s,t)dt$, $a_{01}=-\lambda\int_0^s k(s,t)(t-s)dt$, ..., $a_{nn}=1-\lambda\int_0^s k_s^{(n)}(s,t)\frac{(t-s)^n}{n!}dt$, the errors $x^{(i)}(s)-\bar{x}^{(i)}(s)$ satisfies

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0n} \\ a_{10} & a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & \cdots & a_{nn} \end{bmatrix}\begin{bmatrix} x(s)-\bar{x}(s) \\ x'(s)-\bar{x}'(s) \\ \vdots \\ x^{(n)}(s)-\bar{x}^{(n)}(s) \end{bmatrix}$$
$$=\begin{bmatrix} \frac{x^{(n+1)}(\xi_0(s))}{(n+1)!}\int_0^s k(s,t)(t-s)^{n+1}dt \\ \frac{x^{(n+1)}(\xi_1(s))}{(n+1)!}\int_0^s k_s'(s,t)(t-s)^{n+1}dt \\ \vdots \\ \frac{x^{(n+1)}(\xi_n(s))}{(n+1)!}\int_0^s k_s^{(n)}(s,t)(t-s)^{n+1}dt \end{bmatrix} \tag{2.5}$$

where $s\le T$. Note that, with $\sup_{0\le s,t\le T}|k^{(j)}(s,t)|\le M$, $0\le j\le n$, $M>0$,

$$\left|\int_0^s k(s,t)(t-s)^{n+1}dt\right|\le M|\int_0^s (t-s)^{n+1}dt|\le M\frac{s^{n+2}}{(n+2)!}.$$

Thus

$$\left|\frac{x^{(n+1)}(\xi_1(s))}{(n+1)!}\int_0^s k_s^{(j)}(s,t)(t-s)^{n+1}dt\right|\le CM\frac{T^{n+2}}{(n+2)!}\to 0,\quad \text{as } n\to\infty$$

where $\sup_{0\le s\le T}|x^{(j)}(s)|=C$. This shows that the present Taylor method gives rise to convergent numerical solution, albeit slowly, if s is large. Examples reported in [7] and [8] restrict the values of s to be between 0 and 1. Within this range, the current method provides numerical approximations which are much better than the ones reported in [7] and [8]. Numerical examples are included in this paper where T is taken to be $4,6$ and 8 (see Tables 7, 8 and 9). Tables 7, 8 and 9 demonstrate the error bounds given in (2.5) which are subject to the size of T and that of n. As T increases, a larger n must be chosen for the method to achieve certain accuracy. For example, in Table 7, with $T=4,6$, the degree $n=10$ Taylor expansion gives $O(10^{-3})$ accuracy, but when T is increased to

8, n must be be taken even larger to achieve the same accuracy. On this point, we point out that, as stated in the introduction, the present Taylor method should be viewed as a method which provides accurate approximations to the solution and to its derivatives locally, -i.e., approximation over a relatively small region containing the expansion point.

In matrix form, we denote (2.5) as

$$AE = F.$$

Apparent from the equation (2.5) is that if the solution $x(s)$ is a polynomial of degree n or less, then the current method computes it exactly if computations are carried out exactly. Also, under the assumption that k is sufficiently smooth and for each arbitrary but a fixed s, each term on the right side of (2.5) approaches 0 as $n \to \infty$. Thus, assuming that A^{-1} exists, from (2.5), the error $E = [\varepsilon_i(s)]_{i=0}^n$ with $\varepsilon_i(s) = x^{(i)}(s) - \bar{x}^{(i)}(s)$ satisfies

$$E = A^{-1}F.$$

Hence relative to a matrix norm $\|\cdot\|$, the error vector E satisfies

$$\|E\| \le \|A^{-1}\|\|F\|. \tag{2.6}$$

We now present two examples. These examples are taken from the paper [7]. Numerical results obtained by the current method are consistently better than those reported in [7]. Also, note that the current method also computes the derivatives of the solution simultaneously. The approximation of the first derivative is also given in Table 1. All tables are posted at the end of this paper.

Example 2.1: First, we consider (example 1 of [7]),

$$x(s) - \frac{2}{\pi}\int_0^s k(s,t)x(t)dt = y(s), \qquad 0 < s \le T,$$

where $k(s,t) = [4+(s-t)^2]^{-1}, T = 1$ and $y(s)$ is chosen so that $x(s) = 1+s^2+s^5$ is the solution. Numerical result with $n = 5$ gave excellent approximation for $x(s)$ as well as its derivatives. The numerical approximation for $x(s)$ and its first derivative with $n = 5$ are shown in Table 1. And, the absolute error between exact and approximate value of $x(s)$ with $n = 1, 2, 4$ and 5 are shown in Table 2.

Example 2.2: In this example (example 2 of [7]), the kernel $k(s,t)$ is the same as in Example 2.1 but $y(s)$ is chosen so that $x(s) = e^2 s$ is a solution. Numerical result with $n = 5$ gave excellent approximation for $x(s)$ as well as its derivatives. The numerical approximation for $x(s)$ and its first derivative with $n = 5$ are shown in Table 3. And, the absolute error between exact and approximate value of $x(s)$ with $n = 1, 2, 4$ and 5 are shown in Table 4.

Example 2.3: We consider the following integral equation (example 3 of [7]),

$$x(s) + \int_0^s k(s,t)x(t)dt = 1, \qquad 0 < s \le T,$$

where $k(s,t) = (s-t)$, $T = 1$, and $x(s) = \cos(s)$ is the exact solution. Numerical result with $n = 7$ gave excellent approximation for $x(s)$ as well as its derivatives. The numerical approximation for $x(s)$ and its first derivative with $n = 7$ are shown in Table 5. And, the absolute error between exact and approximate value of $x(s)$ with $n = 4, 5, 6$ and 7 are shown in Table 6.

Example 2.4: Here we consider Examples 2.1-2.3 with $T = 4, 6$ and 8. The numerical error between exact and approximate value of $x(s)$ are shown in Table 7-9, respectively. As stated earlier, as T increases it is necessary for the current Taylor-expansion method to increase n accordingly to maintain the accuracy of the approximation throughout the interval $[0, T]$.

3 System of Volterra Integral Equations: In this section, we apply the method developed in Section 2 to a system of second kind Volterra integral equations. Specifically, we consider

$$\mathbf{X}(s) - \int_0^s \mathbf{K}(s,t)\mathbf{X}(t)dt = \mathbf{Y}(s), \qquad 0 \le s \le T, \tag{3.1}$$

where

$$\begin{aligned} \mathbf{X}(s) &= [x_1(s), x_2(s), \ldots, x_m(s)]^T, \\ \mathbf{Y}(s) &= [y_1(s), y_2(s), \ldots, y_m(s)]^T \\ \mathbf{K}(s,t) &= [k_{ij}(s,t)], \qquad i,j = 1,2,\ldots,m, \end{aligned}$$

and $\mathbf{Y}$, $\mathbf{K}$ are known and $\mathbf{X}$ is the solution to be determined. Now the p th equation of (3.1) is given by

$$x_p(s) - \int_0^s \sum_{q=1}^{m} k_{pq}(s,t)x_q(t)dt = y_p(s), \qquad p = 1,2,\ldots,m. \tag{3.2}$$

Replacing each $x_q(t)$ in (3.2) by

$$x_q(t) \approx x_q(s) + x_q'(s)(t-s) + \cdots + \frac{1}{n!}x_q^{(n)}(s)(t-s)^n \tag{3.3}$$

we obtain

$$x_p(s) - \sum_{q=1}^{m}\sum_{i=0}^{n}\frac{1}{i!}\{\int_0^s k_{pq}(s,t)(t-s)^i dt\}\, x_q^{(i)}(s) \approx y_p(s), \tag{3.4}$$

with $p = 1,2,\ldots,m$.

Following the method explored in Section 2, we now differentiate (3.2) n times and substituting (3.3) into each $x_q(t)$ under the integrals to obtain $m(n+1)$ equations in as many unknowns $x_q^{(i)}(s)$ for $q = 1,2,\ldots,m$, and $i = 0,1,\ldots,n$.

$$\begin{aligned}
x_p'(s) - \{\sum_{q=1}^{m} k_{pq}(s,s)x_q(s) + \sum_{i=0}^{n}\sum_{q=1}^{m}\int_0^s k'_{pq_s}(s,t)\frac{(t-s)^i}{i!}dt x_q^{(i)}(s)\} &= y_p'(s) \\
x_p''(s) - \lambda\{\sum_{q=1}^{m} k_{pq}(s,s)x_q'(s) + \sum_{q=1}^{m} k'_{pq}(s,s)x_q(s) \\
+\sum_{q=1}^{m} k'_{pq_s}(s,s)x_q(s) + \sum_{i=0}^{n}\sum_{q=1}^{m}\int_0^s k''_{pq_s}(s,t)\frac{(t-s)^i}{i!}dt x_p^{(i)}(s)\} &= y''(s) \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
x_p^{(n)}(s) - \lambda\{\sum_{j=0}^{n-1}\binom{n-1}{j}\sum_{q=1}^{m} k_{pq}^{(n-1-j)}(s,s)x_q^{(j)}(s) \\
+\sum_{i=0}^{n-3}\sum_{j=0}^{n-2-i}\binom{n-2-i}{j}[\sum_{q=1}^{m} k_{pq_s}^{(i+1)}(s,s)]^{(n-2-i-j)}x_q^{(j)}(s) \\
+\sum_{i=0}^{n}\sum_{q=1}^{m}\int_0^s k_{pq_s}^{(n)}(s,t)\frac{(t-s)^i}{i!}dt x_q^{(i)}(s)\} &= y^{(n)}(s).
\end{aligned} \tag{3.5}$$

With $m = 2$ and $n = 2$, equations (3.5) can be written as

$$\begin{bmatrix} x_1^{(j)}(s) \\ x_2^{(j)}(s) \end{bmatrix} - \int_0^s \begin{bmatrix} k_{11_s}^{(j)}(s,t) & k_{12_s}^{(j)}(s,t) \\ k_{21_s}^{(j)}(s,t) & k_{22_s}^{(j)}(s,t) \end{bmatrix} \begin{bmatrix} \sum_{i=0}^{2} x_1^{(i)}(s) \dfrac{(t-s)^i}{i!} \\ \sum_{i=0}^{2} x_1^{(i)}(s) \dfrac{(t-s)^i}{i!} \end{bmatrix} dt = \begin{bmatrix} y_1^{(j)}(s) \\ y_2^{(j)}(s) \end{bmatrix}, \quad (3.6)$$

for $j = 0, 1, 2$.

Equations (3.6) represent 6 equations in 6 unknowns $x_1(s), x_1'(s), x_1''(s), x_2(s), x_2'(s)$ and $x_2''(s)$. We once again test the current method on the examples used in the paper [8]. Numerical results obtained are much better than those reported in [8].

Example 3.1: Consider the following Volterra system of integral equation (example 1 of [8]),

$$\begin{cases} x_1(s) - \int_0^s (s-t)^3 x_1(t)dt - \int_0^s (s-t)^2 x_2(t)dt = y_1(s), \\ x_2(s) - \int_0^s (s-t)^4 x_1(t)dt - \int_0^s (s-t)^3 x_2(t)dt = y_2(s). \end{cases}$$

The values of $y_1(s)$ and $y_2(s)$ are chosen so that exact solutions are $x_1(s) = s^2 + 1$ and $x_2(s) = 1 - s^3 + s$. The numerical approximation for $x_1(s)$ and $x_2(s)$ with $n = 4$ are shown in Table 10.

Example 3.2: Consider the following Volterra system of integral equation (example 2 of [8]),

$$\begin{cases} x_1(s) - \int_0^s \big(\sin(s-t) - 1\big) x_1(t)dt - \int_0^s \big(1 - t\cos(s)\big) x_2(t)dt = y_1(s), \\ x_2(s) - \int_0^s x_1(t)dt - \int_0^s (s-t) x_2(t)dt = y_2(s). \end{cases}$$

The values of $y_1(s)$ and $y_2(s)$ are chosen so that exact solutions are $x_1(s) = \cos(s)$ and $x_2(s) = \sin(s)$. The numerical approximation for $x_1(s)$ and $x_2(s)$ with $n = 4$ are shown in Table 11.

REFERENCES

[1] K. E. Atkinson: The Numerical Solution of Integral Equations of the Second Kind, *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge Press, (1997)

[2] Yuhe Ren, Bo Zhang, Hong Qiao: A simple Taylor-series expansion method for a class of second integral equations, *Jl. Comp. and Appl. Math.*, Vol. 110, pp. 15-24, (1999).

[3] P. Huabsomboon, B. Novaprateep, H. Kaneko: On Taylor-series expansion method for the second kind integral equations, *Jl. Comp. and Appl. Math.* , Vol. 234, pp. 1466-1472, (2010).

[4] P. Huabsomboon, B. Novaprateep, H. Kaneko: Taylor-series expansion methods for nonlinear Hammerstein equations, Submitted.

[5] H. Kaneko, Y. Xu: Superconvergence of the Iterated Galerkin Method for Hammerstein Equations, *S.I.A.M. Journal of Numerical Analysis*, Vol. 33, (1996), pp.1048-1064.

[6] K. Maleknejad, N. Aghazadeh, M. Rabbani: Numerical omputational solution of second kind Fredholm integral equations system by using a Taylor-series expansion method, *Appl. Math. and Comp.*, Vol. 175, (2006), pp.1229-1234.

[7] K. Maleknejad, N. Aghazadeh: Numerical solution of Volterra integral equations of the second kind with convolution kernel by using Taylor-series expansion method, *Appl. Math. and Comp.*,Vol. 161, pp 915-922, (2005).

[8] M. Rabbani, K. Maleknejad, N. Aghazadeh: Numerical computational solution of Volterra integral equations system of the second kind by using an expansion method, *Appl. Math. and Comp.*,Vol. 187, pp 1143-1146, (2007).

[1] Department of Mathematics Faculty of Science Mahidol University
Bangkok, 10400, Thailand
[2]Centre of Excellence in Mathematics, PERDO, CHE, THAILAND
[3] Department of Mathematics and Statistics Old Dominion University
Norfolk, VA 23529-0077, USA
[*]Corresponding author

Table 1: Numerical approximation for $x(s)$ and $x'(s)$ in Example 2.1 with $n = 5$

s	$x(s)$			$x'(s)$		
	Exact	Approx.	Error	Exact	Approx.	Error
0.1	1.01001	1.01001	0.00000×10^{0}	0.20050	0.20050	0.00000×10^{0}
0.2	1.04032	1.04032	6.66134×10^{-16}	0.40800	0.40800	1.11022×10^{-16}
0.3	1.09243	1.09243	4.44089×10^{-16}	0.64050	0.64050	0.00000×10^{0}
0.4	1.17024	1.17024	0.00000×10^{0}	0.92800	0.92800	1.11022×10^{-16}
0.5	1.28125	1.28125	2.09388×10^{-13}	1.31250	1.31250	8.43769×10^{-15}
0.6	1.43776	1.43776	3.89577×10^{-12}	1.84800	1.84800	2.78888×10^{-13}
0.7	1.65807	1.65807	9.97045×10^{-11}	2.60050	2.60050	9.84901×10^{-12}
0.8	1.96768	1.96768	1.59121×10^{-10}	3.64800	3.64800	1.87597×10^{-11}
0.9	2.40049	2.40049	3.84105×10^{-8}	5.08050	5.08050	5.86003×10^{-9}
1	3.00000	3.00000	8.20959×10^{-8}	7.00000	7.00000	1.19292×10^{-7}

Table 2: Numerical error between exact and approximate value of $x(s)$ in Example 2.1 with $n = 1, 2, 4$ and 5

s	$n = 1$	$n = 2$	$n = 4$	$n = 5$
0.1	1.28178×10^{-4}	2.69080×10^{-7}	2.69011×10^{-8}	0.00000×10^{0}
0.2	7.61838×10^{-4}	1.74138×10^{-5}	1.73962×10^{-6}	6.66134×10^{-16}
0.3	2.41943×10^{-3}	1.99954×10^{-4}	1.99508×10^{-5}	4.44089×10^{-16}
0.4	6.07989×10^{-3}	1.12919×10^{-3}	1.12478×10^{-4}	0.00000×10^{0}
0.5	1.36930×10^{-2}	4.31745×10^{-3}	4.29138×10^{-4}	2.09388×10^{-13}
0.6	2.88949×10^{-2}	1.28884×10^{-2}	1.27774×10^{-3}	3.89577×10^{-12}
0.7	5.79135×10^{-2}	3.24155×10^{-2}	3.20381×10^{-3}	9.97045×10^{-11}
0.8	1.10644×10^{-1}	7.18919×10^{-2}	7.08020×10^{-3}	1.59121×10^{-10}
0.9	2.01868×10^{-1}	1.44805×10^{-1}	1.42028×10^{-2}	3.84105×10^{-8}
1	3.52597×10^{-1}	2.70300×10^{-1}	2.63893×10^{-2}	8.20959×10^{-8}

Table 3: Numerical approximation for $x(s)$ and $x'(s)$ in Example 2.2 with $n = 5$

s	$x(s)$			$x'(s)$		
	Exact	Approx.	Error	Exact	Approx.	Error
0.1	0.73891	0.73891	0.00000×10^{0}	7.38906	7.38906	8.88178×10^{-16}
0.2	1.47781	1.47781	0.00000×10^{0}	7.38906	7.38906	8.88178×10^{-16}
0.3	2.21672	2.21672	0.00000×10^{0}	7.38906	7.38906	0.00000×10^{0}
0.4	2.95562	2.95562	4.79616×10^{-14}	7.38906	7.38906	0.00000×10^{0}
0.5	3.69453	3.69453	2.91056×10^{-12}	7.38906	7.38906	9.76996×10^{-14}
0.6	4.43343	4.43343	5.48255×10^{-11}	7.38906	7.38906	3.44080×10^{-12}
0.7	5.17234	5.17234	1.01698×10^{-10}	7.38906	7.38906	9.06919×10^{-12}
0.8	5.91124	5.91124	1.12488×10^{-9}	7.38906	7.38906	1.03307×10^{-10}
0.9	6.65015	6.65015	8.20371×10^{-9}	7.38906	7.38906	1.97368×10^{-8}
1	7.38906	7.38906	1.37371×10^{-8}	7.38906	7.38906	1.96379×10^{-8}

Table 4: Numerical error between exact and approximate value of $x(s)$ in Example 2.2 with $n = 1, 2, 4$ and 5

s	$n = 1$	$n = 2$	$n = 4$	$n = 5$
0.1	1.93621×10^{-5}	0.00000×10^{0}	0.00000×10^{0}	0.00000×10^{0}
0.2	6.66505×10^{-4}	0.00000×10^{0}	0.00000×10^{0}	0.00000×10^{0}
0.3	2.82441×10^{-3}	0.00000×10^{0}	0.00000×10^{0}	0.00000×10^{0}
0.4	7.35016×10^{-3}	0.00000×10^{0}	0.00000×10^{0}	4.79616×10^{-14}
0.5	1.50339×10^{-2}	4.44089×10^{-16}	2.92122×10^{-12}	2.91056×10^{-12}
0.6	2.65408×10^{-2}	0.00000×10^{0}	5.12941×10^{-11}	5.48255×10^{-11}
0.7	4.23307×10^{-2}	0.00000×10^{0}	1.04175×10^{-10}	1.01698×10^{-10}
0.8	6.25482×10^{-2}	1.52533×10^{-9}	1.10537×10^{-9}	1.12488×10^{-9}
0.9	8.68808×10^{-2}	6.13235×10^{-9}	8.23474×10^{-9}	8.20371×10^{-9}
1	1.14378×10^{-1}	1.49810×10^{-8}	1.37800×10^{-8}	1.37371×10^{-8}

Table 5: Numerical approximation for $x(s)$ and $x'(s)$ in Example 2.3 with $n = 7$

s	$x(s)$			$x'(s)$		
	Exact	Approx.	Error	Exact	Approx.	Error
0.1	0.99500	0.99500	5.55112×10^{-16}	-0.09983	-0.09983	2.94209×10^{-15}
0.2	0.98007	0.98007	2.34257×10^{-13}	-0.19867	-0.19867	1.39810×10^{-12}
0.3	0.95534	0.95534	1.36007×10^{-11}	-0.29552	-0.29552	5.10094×10^{-11}
0.4	0.92106	0.92106	2.34978×10^{-10}	-0.38942	-0.38942	6.38217×10^{-10}
0.5	0.87758	0.87758	2.09100×10^{-9}	-0.47943	-0.47943	4.39455×10^{-9}
0.6	0.82534	0.82534	1.21844×10^{-8}	-0.56464	-0.56464	2.05580×10^{-8}
0.7	0.76484	0.76484	5.27715×10^{-8}	-0.64422	-0.64422	7.30342×10^{-8}
0.8	0.69671	0.69671	1.83028×10^{-7}	-0.71736	-0.71736	2.10260×10^{-7}
0.9	0.62161	0.62161	5.32773×10^{-7}	-0.78333	-0.78333	5.10466×10^{-7}
1	0.54030	0.54030	1.34194×10^{-6}	-0.84147	-0.84147	1.07086×10^{-6}

Table 6: Numerical error between exact and approximate value of $x(s)$ in Example 2.3 with $n = 4, 5, 6$ and 7

s	$n = 4$	$n = 5$	$n = 6$	$n = 7$
0.1	1.01446×10^{-11}	1.72706×10^{-12}	1.66533×10^{-15}	5.55112×10^{-16}
0.2	2.57840×10^{-9}	4.35530×10^{-10}	1.99707×10^{-12}	2.34257×10^{-13}
0.3	6.52946×10^{-8}	1.08793×10^{-8}	1.13209×10^{-10}	1.36007×10^{-11}
0.4	6.41321×10^{-7}	1.04785×10^{-7}	1.97100×10^{-9}	2.34978×10^{-10}
0.5	3.74032×10^{-6}	5.95507×10^{-7}	1.79145×10^{-8}	2.09100×10^{-9}
0.6	1.56580×10^{-5}	2.41243×10^{-6}	1.07697×10^{-7}	1.21844×10^{-8}
0.7	5.20552×10^{-5}	7.70089×10^{-6}	4.85862×10^{-7}	5.27715×10^{-8}
0.8	1.45970×10^{-4}	2.05502×10^{-5}	1.77356×10^{-6}	1.83028×10^{-7}
0.9	3.58903×10^{-4}	4.75814×10^{-5}	5.49892×10^{-6}	5.32773×10^{-7}
1	7.94456×10^{-4}	9.79251×10^{-5}	1.49681×10^{-5}	1.34194×10^{-6}

Table 7: Numerical error between exact and approximate value of $x(s)$ in Example 2.1 with $T = 4, 6$ and 8.

n	$T = 4$	$T = 6$	$T = 8$
1	7.65507×10^{2}	6.20500×10^{3}	2.53929×10^{4}
2	7.19916×10^{2}	5.93211×10^{3}	2.43160×10^{4}
3	3.37662×10^{2}	2.84762×10^{3}	1.18792×10^{4}
4	6.33579×10^{1}	5.44152×10^{2}	2.31883×10^{3}
5	1.13429×10^{-3}	5.31481×10^{-3}	1.27640×10^{0}
6	8.28725×10^{-4}	7.88018×10^{-3}	1.45067×10^{0}
7	8.61024×10^{-4}	1.14199×10^{-2}	2.15124×10^{0}
8	8.65213×10^{-4}	5.90845×10^{-3}	2.26204×10^{0}
9	8.62170×10^{-4}	4.03571×10^{-3}	2.17851×10^{0}
10	8.49176×10^{-4}	4.89579×10^{-3}	1.57485×10^{0}

Table 8: Numerical error between exact and approximate value of $x(s)$ in Example 2.2 with $T = 4, 6$ and 8.

n	$T = 4$	$T = 6$	$T = 8$
1	1.75226×10^{-5}	9.76309×10^{-6}	5.06619×10^{-5}
2	3.74057×10^{-5}	1.06904×10^{-5}	7.54793×10^{-4}
3	1.92558×10^{-5}	1.07938×10^{-4}	3.95590×10^{-4}
4	4.01123×10^{-5}	7.23217×10^{-5}	4.68446×10^{-5}
5	3.96176×10^{-5}	2.11844×10^{-5}	8.44546×10^{-5}
6	3.36036×10^{-5}	1.72044×10^{-5}	3.15011×10^{-4}
7	3.44448×10^{-5}	8.22698×10^{-5}	1.62327×10^{-3}
8	3.45091×10^{-5}	7.26897×10^{-5}	7.61642×10^{-4}
9	3.46198×10^{-5}	7.18160×10^{-5}	7.94623×10^{-4}
10	3.47224×10^{-5}	7.10469×10^{-5}	1.45000×10^{-4}

Table 9: Numerical error between exact and approximate value of $x(s)$ in Example 2.3 with $T = 4, 6$ and 8.

n	$T = 4$	$T = 6$	$T = 8$
1	3.40211×10^{-1}	1.11613×10^{0}	5.49450×10^{-2}
2	2.80863×10^{-1}	1.02290×10^{0}	1.26721×10^{-1}
3	4.32031×10^{0}	8.46324×10^{-1}	1.77460×10^{-1}
4	2.79263×10^{-1}	7.57364×10^{-1}	1.65546×10^{-1}
5	6.89420×10^{-1}	1.50539×10^{0}	1.14468×10^{-1}
6	9.02918×10^{-2}	1.57743×10^{0}	1.05028×10^{-1}
7	2.83949×10^{-1}	6.60412×10^{-1}	1.95996×10^{-1}
8	9.47514×10^{-2}	3.56844×10^{-1}	2.26230×10^{-1}
9	4.77706×10^{-2}	5.35619×10^{-1}	4.52184×10^{-2}
10	9.41788×10^{-3}	3.46782×10^{0}	3.16329×10^{-1}

Table 10: Numerical approximation for $x_1(s)$ and $x_2(s)$ in Example 3.1 with $n = 4$

s	$x_1(s)$			$x_2(s)$		
	Exact	Approx.	Error	Exact	Approx.	Error
0.1	1.01000	1.01000	0.00000×10^{0}	1.09900	1.09900	0.00000×10^{0}
0.2	1.04000	1.04000	2.22045×10^{-16}	1.19200	1.19200	0.00000×10^{0}
0.3	1.09000	1.09000	2.22045×10^{-16}	1.27300	1.27300	2.22045×10^{-16}
0.4	1.16000	1.16000	0.00000×10^{0}	1.33600	1.33600	2.22045×10^{-16}
0.5	1.25000	1.25000	0.00000×10^{0}	1.37500	1.37500	0.00000×10^{0}
0.6	1.36000	1.36000	3.07184×10^{-9}	1.38400	1.38400	4.39548×10^{-8}
0.7	1.49000	1.49000	4.44089×10^{-16}	1.35700	1.35700	2.22045×10^{-16}
0.8	1.64000	1.64000	2.88074×10^{-9}	1.28800	1.28800	1.84623×10^{-8}
0.9	1.81000	1.81000	4.71085×10^{-9}	1.17100	1.17100	2.22582×10^{-8}
1	2.00000	2.00000	2.18084×10^{-8}	1.00000	1.00000	7.99805×10^{-8}

Table 11: Numerical approximation for $x_1(s)$ and $x_2(s)$ in Example 3.2 with $n = 4$

s	$x_1(s)$			$x_2(s)$		
	Exact	Approx.	Error	Exact	Approx.	Error
0.1	0.99500	0.99500	1.33915×10^{-9}	0.09983	0.09983	1.40215×10^{-10}
0.2	0.98007	0.98007	8.15096×10^{-8}	0.19867	0.19867	1.77274×10^{-8}
0.3	0.95534	0.95534	8.72333×10^{-7}	0.29552	0.29552	2.93156×10^{-7}
0.4	0.92106	0.92107	4.55038×10^{-6}	0.38942	0.38942	2.08238×10^{-6}
0.5	0.87758	0.87760	1.59106×10^{-5}	0.47943	0.47943	9.20678×10^{-6}
0.6	0.82534	0.82538	4.29185×10^{-5}	0.56464	0.56467	2.98011×10^{-5}
0.7	0.76484	0.76494	9.60651×10^{-5}	0.64422	0.64429	7.66561×10^{-5}
0.8	0.69671	0.69689	1.85681×10^{-4}	0.71736	0.71752	1.63264×10^{-4}
0.9	0.62161	0.62193	3.15544×10^{-4}	0.78333	0.78362	2.90750×10^{-4}
1	0.54030	0.54077	4.68306×10^{-4}	0.84147	0.84189	4.16414×10^{-4}