

Article

# Comparison Methods for Solving Non-Linear Sturm–Liouville Eigenvalues Problems

Kamel Al-Khaled \* and Ashwaq Hazaimah

Department of Mathematics and Statistics, Jordan University of Science and Technology, Irbid 22110, Jordan; ashwaq.hazaimah@gmail.com or azalhazaimah17@sci.just.edu.jo

\* Correspondence: kamel@just.edu.jo

Received: 6 June 2020; Accepted: 11 July 2020; Published: 16 July 2020



**Abstract:** In this paper, we present a comparative study between Sinc–Galerkin method and a modified version of the variational iteration method (VIM) to solve non-linear Sturm–Liouville eigenvalue problem. In the Sinc method, the problem under consideration was converted from a non-linear differential equation to a non-linear system of equations, that we were able to solve it via the use of some iterative techniques, like Newton’s method. The other method under consideration is the VIM, where the VIM has been modified through the use of the Laplace transform, and another effective modification has also been made to the VIM by replacing the non-linear term in the integral equation resulting from the use of the well-known VIM with the Adomian’s polynomials. In order to explain the advantages of each method over the other, several issues have been studied, including one that has an application in the field of spectral theory. The results in solutions to these problems, which were included in tables, showed that the improved VIM is better than the Sinc method, while the Sinc method addresses some advantages over the VIM when dealing with singular problems.

**Keywords:** Sinc–Galerkin method; variational iteration method; non-linear sturm–liouville problem; approximate methods

## 1. Introduction

The problem under consideration in this study, is of great importance in mathematics and physics. Many physical issues require an evaluation for the eigenvalues as well as finding the corresponding eigenfunctions to understand the physical interpretation, especially when dealing with vibrations and waves. A general non-linear Sturm–Liouville problem (NSLP) can be written by the following differential equation for a non-linear function  $F$ :

$$\mathbf{L}(y) = \frac{d}{dx} \left[ p(x) \frac{dy(x)}{dx} \right] + [\lambda r(x) - q(x)]F(y(x)) = f(x), \quad x \in (0, 1) \quad (1)$$

subject to the boundary conditions

$$y(0) = 0, \quad y(1) = 0, \quad (2)$$

where the given functions  $p(x) \geq 0, r(x) \geq 0, q(x) \geq 0$  and  $f(x)$  are assumed to be analytic on their domain  $(0, 1)$ . The constants that appear in the above equation are optional. Solving problem (1) with some boundary conditions (2), we obtain non-zero values for the parameter  $\lambda$ , we call them eigenvalues, which exist in infinite number, and characterized by many properties, real, increasing, and simple, and for each eigenvalue there is a corresponding solution, say  $y(x)$ , which known to be eigenfunction, see, [1].

Many previous studies have discussed solutions to the problem (1), especially those with the applications in physics and engineering. When starting an applicable study, such as vibration and stability of deformable bodies issues, it is required to find the eigenvalues first. Engineers are interested in the location of the smallest eigenvalue, as this gives potentially the most visual structure of dynamics systems. The eigenvalues are also crucial in finding the stability region of solutions of NSLP [2]. In [3], the collocation method of the weight residual methods are investigated for the approximate computation of higher SLP. Shannon sampling theory has been used to compute the eigenvalues of regular SLP [1]. Asymptotic formulas for eigenvalues associated with Hills equation have been studied in [4]. In [5], the Sinc–Galerkin method was used to approximate solutions of non-linear problems involving non-linear second, fourth, and sixth-order differential equation. The Sinc–Galerkin method was also used in [6,7] to solve two-point boundary value problem with applications to chemical reactor theory. The authors of [8,9] compared the performance of Sinc–Galerkin methods using Sinc bases for solving linear and non-linear second-order two-point boundary value problem. The VIM was discovered and developed by He [10–12]. Several studies dealt with modification of the method [13,14], and also several authors used the method to solve difficult problems with physical applications [15–17]. In [18] they compute eigenvalues and eigenfunctions of singular two-interval Sturm–Liouville problems. While in [19], they solve fourth order linear differential equations. Fractional Sturm–Liouville problem based on the operational matrix method was presented in [20]. Some existence results for non-linear third-order integro-multi-point boundary value problems are discussed in [21].

In this paper, we introduce two methods for solving (1), (2), Sinc–Galerkin method, and variational iteration method. Stenger [22] originally proposed the numerical solution of ordinary differential equations with the Sinc–Galerkin method. Excellent exposition of the use of Sinc function to approximate differential equations, may be found in [22–24]. A basis element may be transformed to any connected subset of the real line via a composition with a suitable conformal map. In conjunction with the Galerkin method for differential equations, perhaps the most distinctive feature of the basis is its resulting exponential convergence rate of the error,  $\mathcal{O}(\exp(-c\sqrt{N}))$ , where  $c > 0$  and where  $2N + 1$  basis functions are used to build the approximation. Moreover, the convergence rate maintains when the solution of the differential equation has boundary singularities. Of equal practical significance is that the technique’s implementation requires no modification in the presence of singularities. Specifically, the statement of the quadrature, the mesh definition and the resulting matrix structure depend only on the parameters of the differential equation whether it is singular or non-singular.

This paper is organized as follows. The Sinc solution together with the Galerkin method and the development of the scheme is treated in Section 2. We formulate an iterative procedure, and provides a modification of the VIM to solve (1), (2) using VIM in Section 3. Numerical examples which demonstrates the convergence of the Sinc method and compares its performance with VIM are introduced in the last section.

## 2. Sinc Function Approximation

In this part, we will present an overview of some facts and concepts needed to study the topic of Sinc function, and use it in an ideal way to solve the systems in Equation (1), these concepts are taken from the books of Stenger and Lund [22,23].

### 2.1. Sinc Function Properties

It is known that the Sinc function is defined on  $\mathbb{R}$ , and because the period in which the issue is being considered is  $(0, 1)$ , we need to redefine the Sinc function in order to use it during the period and the use of conformal mapping. The Sinc function (known as the band-limited function) defined on the whole real line by

$$\text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0. \end{cases}$$

We denote the equal distances between points by  $h > 0$ , and we define the Sinc translated function as

$$S(k, h)(x) = \text{sinc}\left(\frac{x - kh}{h}\right) = \begin{cases} \frac{\sin\left(\frac{\pi}{h}(x - kh)\right)}{\frac{\pi}{h}(x - kh)}, & x \neq kh \\ 1, & x = kh. \end{cases}, \quad k = 0, \pm 1, \pm 2, \dots$$

If the function  $f$  is defined on all real numbers, and for a positive  $h$ , then we know that the following series, if it is converged, we call it the Whittaker cardinal series of  $f$ .

$$f(x) = C(f, h)(x) = \sum_{k=-\infty}^{\infty} f(kh)S(k, h)(x).$$

**Definition 1.** For  $d > 0$  we define the infinite open strip

$$\mathcal{D}_d = \{z = \xi + i\eta \in \mathbb{C} : |\eta| < d\}. \tag{3}$$

**Definition 2.** Let  $\mathcal{D} \subset \mathbb{C}$  be a simply connected domain with boundary  $\partial\mathcal{D}$  with  $a, b$  denoted as two distinct points on  $\partial\mathcal{D}$ . Let  $\phi$  be a conformal map of  $\mathcal{D}$  onto  $\mathcal{D}_d$  with  $\psi$  as its inverse map. Let  $\Gamma = \{z \in \mathbb{C} : z = \psi(u), u \in \mathbb{R}\}$ . For given  $\phi, \psi$  and a constant  $h$ , set  $z_k = \psi(kh)$  for  $k = 0, \pm 1, \pm 2, \dots$

It is well known that Sinc approximation has exponential error, in order to achieve that, our function under approximation has to satisfy some certain decaying condition, so we introduce the following class of functions denoted by  $B(\mathcal{D})$ , for more details about the terms appeared in the next definitions, readers are recommended to see [22].

**Definition 3.** Let  $B(\mathcal{D})$  be the class of complex-valued functions  $f$  defined on a path-connected domain  $\mathcal{D}$ ;

$$\int_{\psi(t+L)} |f(w)dw| = \mathcal{O}(|t|^a), \quad t \rightarrow \infty$$

where  $a \in [0, 1)$ , and  $L = \{iy : |y| < d\}$ ; and with

$$\mathcal{N}(f; \mathcal{D}) = \liminf_{C \subset \mathcal{D}} \inf_{C \rightarrow \partial\mathcal{D}} \int_C |f(w)dw| < \infty$$

where  $C$  is a simple closed curve in  $\mathcal{D}$ .

The term of exponentially decaying is defined bellow.

**Definition 4.** For  $f \in B(\mathcal{D})$ , the  $f/\phi'$  is exponentially decaying with respect to  $\phi$  if there exist positive constants  $K$  and  $\alpha$  such that

$$\left| \frac{f(t)}{\phi'(t)} \right| \leq K \exp(-\alpha|\phi(t)|), \quad t \in \Gamma. \tag{4}$$

We will use integrals to calculate the residual of functions that belong to class  $B(\mathcal{D})$ . The proof of the following Theorem can be found in [22].

**Theorem 1.** Suppose  $f$  belongs to the class  $B(D)$ . Then we have

$$\left| \int_{\Gamma} f(z) dz - h \sum_{k=-\infty}^{\infty} \frac{f(z_k)}{\phi'(z_k)} \right| \leq \frac{N(f; D)}{1 - e^{-d}} e^{-2\pi d/h}.$$

If  $f/\phi'$  is decaying exponentially with respect to  $\phi$  and belongs to  $B(D)$ , then for  $h = \sqrt{2\pi d/(\alpha N)}$  we get

$$\left| \int_{\Gamma} f(z) dz - h \sum_{k=-N}^N \frac{f(z_k)}{\phi'(z_k)} \right| \leq K e^{-2\pi d\alpha/h}. \quad (5)$$

## 2.2. The General Sinc–Galerkin Method

We will describe the general Galerkin method that is typically used to approximate solution of an operator equation in the form of a linear combination of the elements of a given linearly independent system. To solve the equation

$$\mathbf{L}(y) = f, \quad y(a) = y(b) = 0, \quad (6)$$

**we assume** a suitable Hilbert space  $\mathbf{H}$  and  $y \in \mathbf{H}$ . Let  $\{\chi_i\}$  be a dense linearly independent set in  $\mathbf{H}$  and let  $\mathbf{H}^N = \text{span}\{\chi_1, \dots, \chi_N\}$ . For the solution  $y \in \mathbf{H}$  we define the approximation  $y_N \in \mathbf{H}^N$  as a linear combination

$$y_N(x) = \sum_{j=1}^N y_j \chi_j(x) \quad (7)$$

with  $\chi_j = S(j, h) \circ \phi(x)$ . The coefficients  $\{y_j\}$  are to be determined. Galerkin's idea is to require that residual

$$R_N(x) = (\mathbf{L}(y_N) - f)(x),$$

is orthogonal to each  $\chi_j \in \mathbf{H}^N$  by the inner product in  $\mathbf{H}$ :

$$\langle R_N, \chi_j \rangle = 0, \quad 1 \leq j \leq N.$$

In matrix form:

$$\sum_{k=1}^N \langle \mathbf{L}\chi_k, \chi_j \rangle y_k = \langle f, \chi_j \rangle, \quad 1 \leq j \leq N.$$

These equations determine all  $\{y_k\}$ . Taking  $\chi_i$  to be Dirac delta functions, we see that Sinc collocation methods are special case of Sinc-Galerkin methods. Galerkin's method is a powerful tool not only for finding approximate solutions, but also for proving existence theorems of solutions of linear and non-linear equations, especially in problems involving partial differential equations.

## 2.3. The Sinc Methodology

To solve the differential equation under consideration (1) using the Sinc methodology for all  $x$  in  $(0, 1)$ , and since the Sinc function is defined and build on  $\mathbb{R}$ , we have to make some kind of adaptation of the Sinc via conformal mapping. Suppose that function  $f \in C^\infty(\mathbb{R})$  is approximable on  $\mathbb{R}$ , and it satisfies certain smoothness and boundedness conditions on a  $\mathcal{D}$  that contains  $\mathbb{R}$ . In this manner, we are dealing with the "eye-shaped" region (see Figure 1.7, p. 68, of [22]) that contains our domain  $(0, 1)$ . Therefore, to start our approximation on the domain  $(0, 1)$ , we define the eye-shaped domain  $\mathcal{D}_E$  in the  $z$ -plane:

$$\mathcal{D}_E = \{z = x + iy \in \mathbb{C} : |\arg(z/(1-z))| < d \leq \pi/2\}. \quad (8)$$

The eye-shaped domain  $\mathcal{D}_E$  is mapped conformally onto the infinite strip  $\mathcal{D}_d$  via

$$\phi(z) = \ln\left(\frac{z}{1-z}\right),$$

The conformal map  $\phi(z)$  maps  $\mathcal{D}_E$  onto  $\mathcal{D}_d$ , appropriate for using Sinc methodology to approximate solution of (1) with boundary conditions (2). According to constructing the new basis functions defined on  $(0, 1)$  as two functions compositions:

$$S_j(z) = S(j, h) \circ \phi(z) = \text{sinc} \left( \frac{\phi(z) - jh}{h} \right), z \in \mathcal{D}_E.$$

To go back to original domain, we use the inverse of  $u = \phi(z)$ :

$$z = \phi^{-1}(u) = \psi(u) = \frac{e^u}{1 + e^u}.$$

For the equally interpolation points  $\{jk\}_{j=-\infty}^{\infty}$ , we need to define the inverse images of the Sinc domain  $\mathbb{R}$  as

$$\Gamma = \{\psi(t) \in \mathcal{D}_E : -\infty < t < \infty\} = (0, 1),$$

with

$$x_j = \psi(jh) = \frac{e^{jh}}{1 + e^{jh}}, j = 0, \mp 1, \mp 2, \dots$$

respectively. Discrete system for Equation (1) is obtain by rewriting Equation (1) as

$$y''(x) + \tau(x)y'(x) + \lambda R(x)y(x) + Q(x)F(y(x)) = K(x), \tag{9}$$

where  $\tau(x) = p(x)/p'(x)$ ,  $R(x) = r(x)/p(x)$ ,  $Q(x) = q(x)/p(x)$ , and  $K(x) = f(x)/p(x)$ . In order to establish discrete system for (9), we replace  $y(x)$  and its derivatives by

$$y_N(x) = \sum_{j=-N}^N y_j S_j \circ \phi(x) \tag{10}$$

The  $y_j$  are constants appeared in (9) and they are specified later. Recall that

$$R_N = \mathbf{L}(y_N) - K \tag{11}$$

and suppose that metric internally product  $\langle \cdot, \cdot \rangle$  is possessed as

$$\langle f, g \rangle = \int_0^1 f(x)g(x)w(x)dx. \tag{12}$$

Here  $w(x)$  performs a weight function, selected for different cases. Although other logical reasoning exists, the option made here was a consequence of the request that boundary conditions at end points are zero. It is appropriate to pick  $w(x) = 1/\phi'(x)$ . As a perfect debate on the options of the weight functions, we recommend a reader to see [22,23]. Orthogonalizing the remainder with regards to

$$\vec{S} = \begin{bmatrix} S_{-N} \circ \phi \\ S_{-N+1} \circ \phi \\ \vdots \\ S_N \circ \phi \end{bmatrix} \text{ and } \vec{S}w = \begin{bmatrix} S_{-N} \circ \frac{\phi}{\phi'} \\ S_{-N+1} \circ \frac{\phi}{\phi'} \\ \vdots \\ S_N \circ \frac{\phi}{\phi'} \end{bmatrix} \tag{13}$$

yields the framework

$$\langle y_N'', \vec{S} \rangle + \langle \tau(x)y_N', \vec{S} \rangle + \langle \lambda R(x)y_N, \vec{S} \rangle + \langle Q(x)F(y_N), \vec{S} \rangle = \langle K, \vec{S} \rangle \tag{14}$$

Straight away we exchange  $y_N$  by  $y$ . As a subsequent of the criterion for the Sinc–Galerkin procedure, the first expression in Equation (14) is integrated by parts twice and the latter expression once. Postulating the condition of vanishing at both end of the interval,  $y(0) = 0 = y(1)$  Equation (14) reduces to

$$\int_0^1 y(x)[\vec{S}w]''(x)dx - \int_0^1 y(x)[\tau\vec{S}w]'(x)dx + \int_0^1 y(x)[\lambda R\vec{S}w](x)dx + \int_0^1 F(y(x))[Q\vec{S}w](x)dx \\ = \int_0^1 y(x)[K\vec{S}w](x)dx. \quad (15)$$

To build an approximate solution by means of the Sinc–Galerkin method, we must assess the integrals in (15) obtaining a linearly framework solution. The standard Sinc quadrature formula (24) will be exercised. Specifics of the quadrature formula and situations controlling its error bounds (see [22]). To integrate a function  $G(x)$  defined on  $(0, 1)$  and fulfill the postulates of the quadrature formula, subsequently for  $x_k = \phi^{-1}(kh)$  we arrived at

$$\left| \int_0^1 G(x)dx - h \sum_{k=-N}^N \frac{G(x_k)}{\phi'(x_k)} \right| \leq \mathcal{O}(e^{-\kappa\sqrt{N}}).$$

The Sinc–Galerkin method seek to know the derivatives of composite Sinc functions predestined at the nodes. For a conformal map  $\phi$  of  $\mathcal{D}_E$  onto  $\mathcal{D}_d$ , we use the next codes

$$\delta_{i,j}^{(0)} \equiv \begin{cases} 1, & i = j \\ 0, & i \neq j, \end{cases} \quad (16)$$

$$\delta_{i,j}^{(1)} \equiv h \frac{d}{d\phi} [S_i \circ \phi(x)] \Big|_{x=x_j} = \begin{cases} 0, & i = j \\ \frac{(-1)^{j-i}}{j-i}, & i \neq j, \end{cases} \quad (17)$$

and,

$$\delta_{i,j}^{(2)} \equiv h^2 \frac{d^2}{d\phi^2} [S_i \circ \phi(x)] \Big|_{x=x_j} = \begin{cases} \frac{-\pi^2}{3}, & i = j \\ \frac{-2(-1)^{j-i}}{(j-i)^2}, & i \neq j. \end{cases} \quad (18)$$

Thus, the expressions in the  $i$ -th equation of (15) are approximated by

$$\int_0^1 y(x)[S_i \circ \phi w]''(x)dx \approx h \sum_{j=-N}^N \left[ \frac{1}{h^2} \delta_{i,j}^{(2)} \phi'(x_j) w(x_j) + \frac{1}{h} \delta_{i,j}^{(1)} \left( \frac{\phi''(x_j)}{\phi'(x_j)} w(x_j) + 2w'(x_j) \right) \right. \\ \left. + \delta_{i,j}^{(0)} \frac{w''(x_j)}{\phi'(x_j)} \right] y(x_j)$$

the second integral in Equation (15) can be written as,

$$\int_0^1 y(x)[\tau S_i \circ \phi w]'(x)dx \approx h \sum_{j=-N}^N \left[ \frac{1}{h} \delta_{i,j}^{(1)} \tau(x_j) w(x_j) + \delta_{i,j}^{(0)} \frac{(\tau w)'(x_j)}{\phi'(x_j)} \right] y(x_j)$$

while the third integral in Equation (15) as,

$$\int_0^1 y(x)[\lambda R S_i \circ \phi w](x) dx \approx h \lambda \sum_{j=-N}^N \delta_{i,j}^{(0)} R(x_j) \frac{w(x_j)}{\phi'(x_j)} y(x_j)$$

Finally, the last integral on the left-hand side of Equation (15) can be approximated by the finite sum

$$\int_0^1 y(x)[Q S_i \circ \phi w](x) dx \approx h \sum_{j=-N}^N \delta_{i,j}^{(0)} Q(x_j) \frac{w(x_j)}{\phi'(x_j)} F(y(x_j))$$

while the integral on the right-hand side of Equation (15) has the form

$$\int_0^1 K(x)[S_i \circ \phi w](x) dx \approx h \sum_{j=-N}^N \delta_{i,j}^{(0)} \frac{w(x_j)}{\phi'(x_j)} K(x_j)$$

The next coding will be useful to construct the system. Call the  $(2N + 1) \times (2N + 1)$  matrices

$$I^{(p)} = [\delta_{i,j}^{(p)}], p = 1, 2. \tag{19}$$

For instant, the  $i, j$ th components in matrix  $I^{(2)}$  are as in (18). Denote  $D(y) = \text{diag}[y(z_{-N}), \dots, y(z_N)]^T$ , where the superscript “ $T$ ” indicate the transpose of a matrix. The discrete system for using the Sinc–Galerkin system coincide with (15) and can be written in a further, convenient exemplification:

$$\left( U + V + \lambda D(R)D\left(\frac{w}{\phi'}\right) + D(Q)D\left(\frac{w}{\phi'}\right) \right) \vec{Y}_{2N+1} = D\left(\frac{w}{\phi'}\right) \vec{F}_{2N+1} \tag{20}$$

Matrix  $U$  represents the Sinc approximation of the second derivative of  $y$ , written as

$$U = \left[ \frac{1}{h^2} I^{(2)} D(\phi' w) + \frac{1}{h} I^{(1)} D\left(\frac{\phi''}{\phi'} w + 2w'\right) + D\left(\frac{w''}{\phi'}\right) \right] \tag{21}$$

while:

$$V = - \left[ \left( \frac{1}{h} I^{(1)} D(w) + D\left(\frac{w'}{\phi'}\right) \right) \mathcal{D}(\tau) + D\left(\frac{w}{\phi'}\right) D(\tau') \right] \tag{22}$$

$$\vec{F}(y(x))_{2N+1} = \begin{bmatrix} F(y_{-N}) \\ F(y_{-N+1}) \\ \vdots \\ F(y_N) \end{bmatrix}, \quad \vec{K}_{2N+1} = \begin{bmatrix} K(x_{-N}) \\ K(x_{-N+1}) \\ \vdots \\ K(x_N) \end{bmatrix} \tag{23}$$

With this view, we have reached the fact

**Theorem 2.** *The discrete solution for the non-linear SLP via the use of Sinc–Galerkin method for the locating the constants  $\{y_j\}, j = -N, \dots, N$  is given by*

$$\left( U + V + \lambda D(R)D\left(\frac{w}{\phi'}\right) + D(Q)D\left(\frac{w}{\phi'}\right) \right) \vec{Y}_{2N+1} = D\left(\frac{w}{\phi'}\right) \vec{F}_{2N+1}$$

As a conclusion, we end up with a non-linear system of  $(2N + 1)$  equations in  $(2N + 1)$  unknown constants  $\{y_j\}_{j=-N}^N$ . The non-linear system can be solved via the use of Newton’s method. The values found for  $\{y_j\}_{j=-N}^N$  give raise of the approximate Sinc solution  $y_T(x)$  for  $y(x)$ .

#### 2.4. Newton's Method

Here we rewrite the system of non-linear Equation (20) in the form:

$$\mathbf{M}(\mathbf{y}) = \begin{bmatrix} M_{-N}(y_{-N}, y_{-N+1}, \dots, y_N) \\ M_{-N+1}(y_{-N}, y_{-N+1}, \dots, y_N) \\ \vdots \\ M_N(y_{-N}, y_{-N+1}, \dots, y_N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (24)$$

An appropriate method for solving (24) is Newton's method, due to nonlinearity terms appeared in the system. Newton's method suggests starting with an initial guess for the unknown function, an iteration  $i$ . Let  $\mathbf{y}^{(i)}$  be the guess. Let  $\mathbf{M}^{(i)}$  denotes the value of  $\mathbf{M}$  evaluated at the  $i$ th iteration. If the norm of  $\mathbf{M}^{(i)}$  is small enough, we are looking for updates for vectors  $\Delta\mathbf{y}^{(i)}$ , where  $\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}$ , which can be written in components as

$$\begin{bmatrix} y_{-N}^{(i+1)} \\ y_{-N+1}^{(i+1)} \\ \vdots \\ y_N^{(i+1)} \end{bmatrix} = \begin{bmatrix} y_{-N}^{(i)} \\ y_{-N+1}^{(i)} \\ \vdots \\ y_N^{(i)} \end{bmatrix} + \begin{bmatrix} \Delta y_{-N}^{(i)} \\ \Delta y_{-N+1}^{(i)} \\ \vdots \\ \Delta y_N^{(i)} \end{bmatrix}$$

with the goal of reaching  $\mathbf{M}(\mathbf{y}^{(i+1)}) = 0$ . Taylor's extension theorem for functions  $\mathbf{M} \in \mathbb{R}^{2N+1}$  can be used to approximate  $\mathbf{M}(\mathbf{y})$  that is close to  $\mathbf{y}^{(i)}$  as

$$\mathbf{M}(\mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}) = \mathbf{M}(\mathbf{y}^{(i)}) + \nabla\mathbf{M}(\mathbf{y}^{(i)})\Delta(\mathbf{y}^{(i)}) + \mathcal{O}(\|\Delta\mathbf{y}^{(i)}\|^2)$$

where  $\nabla\mathbf{M}(\mathbf{y}^{(i)})$  to denotes the Jacobian matrix given by given by

$$\nabla\mathbf{M}(\mathbf{y}) = \begin{bmatrix} \frac{\partial M_{-N}}{\partial y_{-N}}(\mathbf{y}) & \frac{\partial M_{-N}}{\partial y_{-N+1}}(\mathbf{y}) & \dots & \frac{\partial M_{-N}}{\partial y_N}(\mathbf{y}) \\ \frac{\partial M_{-N+1}}{\partial y_{-N}}(\mathbf{y}) & \frac{\partial M_{-N+1}}{\partial y_{-N+1}}(\mathbf{y}) & \dots & \frac{\partial M_{-N+1}}{\partial y_N}(\mathbf{y}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial M_N}{\partial y_{-N}}(\mathbf{y}) & \frac{\partial M_N}{\partial y_{-N+1}}(\mathbf{y}) & \dots & \frac{\partial M_N}{\partial y_N}(\mathbf{y}) \end{bmatrix}.$$

Ignoring the higher order terms and evaluating the Jacobian  $\nabla\mathbf{M}(\mathbf{y})$  at  $\mathbf{y}^{(i)}$ , we arrive at

$$\mathbf{M}(\mathbf{y}^{(i)} + \Delta\mathbf{y}^{(i)}) = \mathbf{M}(\mathbf{y}^{(i)}) + \nabla\mathbf{M}(\mathbf{y}^{(i)})\Delta(\mathbf{y}^{(i)}) = 0,$$

or

$$\nabla\mathbf{M}\mathbf{y}^{(i)}\Delta\mathbf{y}^{(i)} = -\mathbf{M}(\mathbf{y}^{(i)}).$$

The equation above is a system of  $2N + 1$  linear equations in  $2N + 1$  unknowns  $\Delta\mathbf{y}^{(i)}$ . We stop Newton's iteration whenever  $\|\mathbf{y}^{(i+1)} - \mathbf{y}^{(i)}\| \leq \epsilon$  for given  $\epsilon$ . We also would like to mention that other iterative techniques can be used to solve our non-linear systems (20) such as secant method and quasi-Newton's method.



### 3. The Variational Iteration Method

The VIM proposed by Ji-Huan He [10,11] is an analytical method based on Lagrange multipliers. Powerful, easy and effective method for solving large classes of linear and non-linear problems. For linear problems, the exact solution can be evaluated by only one iteration step because the Lagrange multiplier can be exactly identified. It needs no discretization and no perturbations unlike the traditional methods.

#### 3.1. Analysis of the VIM

To illustrate the basic concepts of He's VIM, we consider the system

$$Ty(x) = g(x), \quad x \in I. \quad (25)$$

Here  $T$  is differential operator that acts on the function  $y$  defined on an interval  $I \subseteq \mathbb{R}$  and  $g$  is a known analytic function defined for all  $x \in \mathbb{R}$ .

The VIM is based on dividing  $T$  into linear and non-linear operators:

$$Ly(x) + Ny(x) = g(x). \quad (26)$$

Here  $L$  is linear,  $N$  is non-linear differential operator and  $g(x)$  is the inhomogeneous term of the equation. The correction functional of Equation (26) can be constructed as follows

$$y_{n+1}(x) = y_n(x) + \int_0^x \mu \{Ly_n(s) + N\tilde{y}_n(s) - g(s)\} ds. \quad (27)$$

Here  $\mu$  is the Lagrange multiplier, which can be identified optimally by variational theory,  $y_n$  are the  $n^{\text{th}}$  order approximate solutions and the  $\tilde{y}_n$  are their restricted variations with their own variations  $\delta\tilde{y}_n = 0$ . The successive approximation  $y_n$ ,  $n \geq 1$  will be readily obtained upon using the determined  $\mu$  and by using any selective function  $y_0$ . Consequently, the solution is given by

$$y(x) = \lim_{n \rightarrow \infty} y_n(x),$$

where  $y_n$  has a limit as  $n \rightarrow \infty$ . The convergent proof for the series solution can be found in [10]. To increase the convergence rate of the truncated series solution derived from VIM, we do a simple modification, the Laplace VIM (LVIM), through the following steps (see, [25])

1. convert the truncated series obtained by VIM by using Laplace transform.
2. approximate the result that we get from previous step using the Padé approximant.
3. convert the output function that we get from the previous step using inverse Laplace transform.

In the next subsection, we introduce our modification for the VIM, where the main idea is to combine Laplace transform and Adomian polynomials to the original VIM. For more details see [25].

#### 3.2. Adomian—Variational Iteration Method (AVIM)

Considering the non-linear operator  $F(y(x))$ , the first few Adomian polynomials are given by

$$\begin{aligned} A_0 &= F(y_0) \\ A_1 &= y_1 F'(y_0) \\ A_2 &= y_2 F'(y_0) + \frac{1}{2!} y_1^2 F''(y_0) \\ A_3 &= y_3 F'(y_0) + y_1 y_2 F''(y_0) + \frac{1}{3!} y_1^3 F'''(y_0). \\ &\vdots \end{aligned} \quad (28)$$

The basic idea of VIM linked with Adomian polynomials is to approximate the general type of differential Equation (1). The main idea of our modified method is to construct the correction functional via the use of VIM. The non-linear term  $Ny(x)$  is replaced by Adomian polynomials ([26]) as

$$Ny(x) = \sum_{k=0}^{\infty} A_k(x), \quad (29)$$

where the first few Adomian polynomials for a non-linear function  $F$  is given as above. Therefore, our approximate solution can be written in the form

$$y_{n+1}(x) = y_n(x) + \int_0^x \mu \{ Ly_n(s) + \sum_{k=0}^n A_k(s) - g(s) \} ds. \quad (30)$$

Upon choosing the initial approximation  $y_0$  as arbitrary, our first approximate solution has the form

$$y_1(x) = y_0(x) + \int_0^x \mu \{ Ly_0(s) + A_0(s) - g(s) \} ds. \quad (31)$$

And

$$y_2(x) = y_1(x) + \int_0^x \mu \{ Ly_1(s) + (A_0(s) + A_1(s)) - g(s) \} ds. \quad (32)$$

And so on, we may compute terms as much as we need to reach sufficient good approximation. Some examples will be given to illustrate the idea. For more details on using Adomian's techniques in finding eigenvalues see [27].

### 3.3. Lagrange Multiplier for Special Kind of Equations

Consider the following ODE

$$y''(x) + \frac{h'(x)}{h(x)} y'(x) + f(t, y(x)) = g(x), y(0) = A, y(1) = B, \quad (33)$$

where  $f(t, y(x))$  and  $g(x)$  are continuous real-valued functions and  $h(x)$  is a continuous and differentiable function with  $h(x) \neq 0$ . Bratu, Emden–Fowler, LaneEmden, Poisson–Boltzmann Lagerstrom and more other equations are special cases of (33). For solving Equation (33) by VIM, we construct the correction functional as follows

$$y_{n+1}(x) = y_n(x) + \int_0^x \lambda(x, s) \{ y''(s) + \frac{h'(s)}{h(s)} y'(s) + f(s, y(s)) - g(s) \} ds. \quad (34)$$

Making the above correction functional stationary with respect to  $y_n$  and noticing that  $\delta y_n(0) = 0$ , it yields

$$\begin{aligned} \delta y_{n+1}(x) &= \delta y_n(x) + \delta \int_0^x \lambda(x, s) \{ y''(s) + \frac{h'(s)}{h(s)} y'(s) + f(s, u(s)) - g(s) \} ds. \\ &= \delta y_n(x) + \left( \lambda(x, s) \frac{h'(s)}{h(s)} \delta y_n(s) + \lambda(t, s) \delta y'_n(s) + \frac{\partial \lambda(x, s)}{\partial s} \delta y_n(s) \right) \Big|_{s=x} \\ &\quad + \int_0^x \left\{ \frac{\partial^2 \lambda(x, s)}{\partial s^2} + \frac{\partial}{\partial s} \left[ \lambda(x, s) \frac{h'(s)}{h(s)} \delta y_n(s) \right] \right\} ds. \\ &= 0. \end{aligned} \quad (35)$$

Therefore, the following stationary conditions are obtained as

$$\frac{\partial^2 \lambda(x, s)}{\partial s^2} + \frac{\partial}{\partial s} \left[ \lambda(x, s) \frac{h'(x)}{h(x)} \delta y_n(x) \right] = 0 \quad (36)$$

$$1 + \lambda(x, x) \frac{h'(s)}{h(s)} - \frac{\partial \lambda(x, s)}{\partial s} \Big|_{s=x} \quad (37)$$

$$\lambda(x, x) = 0 \quad (38)$$

Therefore, the Lagrange multiplier can be readily identified with

$$\lambda(x, s) = h(s) \left[ \int \frac{ds}{h(s)} - \int \frac{dx}{h(x)} \right]. \quad (39)$$

#### 4. Numerical Results

To show the efficiency of the methods described in the previous sections, we present two examples. The first example is non-linear, and will be tested using VIM and their modifications appeared in Sections 3.1 and 3.2. A Sinc solution will also be provided. The second example is chosen to deal with finding eigenvalues for Titchmarsh equation and some tabulated numerical results are demonstrated. In the Sinc–Galerkin method, we choose  $d = \pi/2$  and  $\alpha = 1/2$ . The step size and the summation limit  $N$  are selected, therefore the error is asymptotically balanced. Once  $N$  is chosen, the step size is determined by  $h = \sqrt{\frac{\pi d}{\alpha N}}$ . In using Newton's iterative technique, we start with a zero initial guess, and we stop criterion when we reach  $\| \mathbf{y}^{(i+1)} - \mathbf{y}^{(i)} \| \leq 10^{-7}$ .

**Example 1.** We consider a version of the Duffing equation [25]

$$y'' + 2y' + y + 8y^3 - e^{-3x} = 0. \quad (40)$$

With conditions

$$y(0) = 0.5 \text{ and } y(1) = \frac{1}{2e} \quad (41)$$

It is known that exact solution is given with  $y(x) = 0.5e^{-x}$ .

For the VIM solution, we start with the functional correction in (40) written as

$$y_{n+1}(x) = y_n(x) + \int_0^x \lambda(x, s) \{ y''(s) + 2y'(s) + y(s) + 8y^3(s) - e^{-s} \} ds. \quad (42)$$

By (39), the Lagrange multiplier is identified as

$$\lambda(x, s) = h(s) \left( \int \frac{ds}{h(s)} - \int \frac{dx}{h(x)} \right). \quad (43)$$

In our case,

$$\frac{h'(x)}{h(x)} = 2, \text{ so } h = e^{2x}. \quad (44)$$

Using (43), we arrived at

$$\lambda(x, s) = \frac{1}{2} \left( e^{2s-2x} - 1 \right). \quad (45)$$

We apply the LVIM, mentioned at the end of Section 3.1, to obtain the following iteration formula:

$$y_{n+1}(x) = y_n(x) + \int_0^x \frac{1}{2} (e^{2s-2x} - 1) \{y''(s) + 2y'(s) + y(s) + 8y^3(s) - e^{-s}\} ds. \quad (46)$$

Given that the initial guess is chosen arbitrarily, we assume  $y_0 = A + Bx$ , so our first approximation is given by

$$\begin{aligned} y_1(x) = & -4A^3x - 2A^3e^{-2x} + 2A^3 - 6A^2Bx^2 + 6A^2Bx + 3A^2Be^{-2x} - 3A^2B - 4AB^2x^3 + 6AB^2x^2 \\ & - 6AB^2x - 3AB^2e^{-2x} + 3AB^2 - \frac{Ax}{2} - \frac{1}{4}Ae^{-2x} + \frac{5A}{4} - B^3x^4 + 2B^3x^3 - 3B^3x^2 + 3B^3x \\ & + \frac{3}{2}B^3e^{-2x} - \frac{3B^3}{2} - \frac{Bx^2}{4} + \frac{Bx}{4} - \frac{3}{8}Be^{-2x} + \frac{3B}{8} + \frac{e^{-3x}}{3} - \frac{e^{-2x}}{2} + \frac{1}{6}. \end{aligned} \quad (47)$$

We use the conditions appeared in Equation (41) to get  $A = -0.5$  and  $B = -0.5$ . In the same manner, using the obtained values  $A, B$ , the second iteration is given by

$$\begin{aligned} y_2(x) = & -0.000600962x^{13} + 0.015625x^{12} - 0.204545x^{11} + 1.79531x^{10} - 0.0338542e^{-2.x}x^9 \\ & - 11.8681x^9 - 0.0416667e^{-3.x}x^8 + 0.304688e^{-2.x}x^8 + 62.793x^8 + 0.0555556e^{-3.x}x^7 - 1.74107e^{-2.x}x^7 \\ & - 274.77x^7 - 1.53704e^{-3.x}x^6 + 5.89063e^{-2.x}x^6 + 1008.25x^6 - 3.24074e^{-3.x}x^5 - 15.7625e^{-2.x}x^5 - 3096.97x^5 \\ & - 0.0138889e^{-6.x}x^4 + 0.216667e^{-5.x}x^4 - 0.990234e^{-4.x}x^4 - 29.0957e^{-3.x}x^4 + 27.3203e^{-2.x}x^4 \\ & + 7829.67x^4 + 0.0601852e^{-6.x}x^3 - 0.837778e^{-5.x}x^3 + 2.9707e^{-4.x}x^3 - 97.072e^{-3.x}x^3 - 41.099e^{-2.x}x^3 \\ & - 15739.5x^3 - 0.140046e^{-6.x}x^2 + 1.95289e^{-5.x}x^2 - 7.67432e^{-4.x}x^2 - 311.183e^{-3.x}x^2 + 34.4297e^{-2.x}x^2 \\ & + 23663.x^2 + 0.173804e^{-6.x}x - 2.34847e^{-5.x}x + 8.04565e^{-4.x}x - 609.4e^{-3.x}x - 28.9792e^{-2.x}x \\ & - 23686.9x - 0.00470312e^{-9.x} + 0.0902778e^{-8.x} - 0.603571e^{-7.x} + 1.31535e^{-6.x} + 1.59265e^{-5.x} \\ & - 6.24054e^{-4.x} + \frac{e^{-3x}}{3} - 613.749e^{-3.x} - 1.625e^{-2x} - 11231.4e^{-2.x} + 11850.8. \end{aligned} \quad (48)$$

Applying Laplace transform to both sides of Equation (47), we get

$$\begin{aligned} \mathcal{L}[y_1(x)](s) = & -\frac{3.7422 \times 10^6}{s^{14}} + \frac{7.4844 \times 10^6}{s^{13}} - \frac{8.1648 \times 10^6}{s^{12}} + \frac{6.51483 \times 10^6}{s^{11}} - \frac{4.30668 \times 10^6}{s^{10}} \\ & + \frac{2.53181 \times 10^6}{s^9} - \frac{1.38484 \times 10^6}{s^8} + \frac{725942.}{s^7} - \frac{371637.}{s^6} + \frac{187912.}{s^5} - \frac{94437.3}{s^4} + \frac{47325.9}{s^3} - \frac{23686.9}{s^2} \\ & - \frac{1.625}{s+2} - \frac{11231.4}{s+2.} - \frac{28.9792}{(s+2.)^2} + \frac{68.8594}{(s+2.)^3} - \frac{246.594}{(s+2.)^4} + \frac{655.688}{(s+2.)^5} - \frac{1891.5}{(s+2.)^6} + \frac{4241.25}{(s+2.)^7} - \frac{8775.}{(s+2.)^8} \\ & + \frac{12285.}{(s+2.)^9} - \frac{12285.}{(s+2.)^{10}} + \frac{1}{3(s+3)} - \frac{613.749}{s+3.} - \frac{609.4}{(s+3.)^2} - \frac{622.366}{(s+3.)^3} - \frac{582.432}{(s+3.)^4} - \frac{698.296}{(s+3.)^5} - \frac{388.889}{(s+3.)^6} \\ & - \frac{1106.67}{(s+3.)^7} + \frac{280.}{(s+3.)^8} - \frac{1680.}{(s+3.)^9} - \frac{6.24054}{s+4.} + \frac{8.04565}{(s+4.)^2} - \frac{15.3486}{(s+4.)^3} + \frac{17.8242}{(s+4.)^4} - \frac{23.7656}{(s+4.)^5} + \frac{1.59265}{s+5.} \\ & - \frac{2.34847}{(s+5.)^2} + \frac{3.90578}{(s+5.)^3} - \frac{5.02667}{(s+5.)^4} + \frac{5.2}{(s+5.)^5} + \frac{1.31535}{s+6.} + \frac{0.173804}{(s+6.)^2} - \frac{0.280093}{(s+6.)^3} + \frac{0.361111}{(s+6.)^4} - \frac{0.333333}{(s+6.)^5} \\ & - \frac{0.603571}{s+7.} + \frac{0.0902778}{s+8.} - \frac{0.00470312}{s+9.} + \frac{11850.8}{s}. \end{aligned} \quad (49)$$

To simplify the matter, we replace  $s = \frac{1}{t}$

$$\begin{aligned} \mathcal{L}[y_1(x)]\left(\frac{1}{t}\right) = & -\frac{11231.4}{2. + \frac{1}{t}} - \frac{28.9792}{\left(2. + \frac{1}{t}\right)^2} + \frac{68.8594}{\left(2. + \frac{1}{t}\right)^3} - \frac{246.594}{\left(2. + \frac{1}{t}\right)^4} + \frac{655.688}{\left(2. + \frac{1}{t}\right)^5} - \frac{1891.5}{\left(2. + \frac{1}{t}\right)^6} + \frac{4241.25}{\left(2. + \frac{1}{t}\right)^7} \\ & - \frac{8775.}{\left(2. + \frac{1}{t}\right)^8} + \frac{12285.}{\left(2. + \frac{1}{t}\right)^9} - \frac{12285.}{\left(2. + \frac{1}{t}\right)^{10}} - \frac{613.749}{3. + \frac{1}{t}} - \frac{609.4}{\left(3. + \frac{1}{t}\right)^2} - \frac{622.366}{\left(3. + \frac{1}{t}\right)^3} - \frac{582.432}{\left(3. + \frac{1}{t}\right)^4} - \frac{698.296}{\left(3. + \frac{1}{t}\right)^5} \\ & - \frac{388.889}{\left(3. + \frac{1}{t}\right)^6} - \frac{1106.67}{\left(3. + \frac{1}{t}\right)^7} + \frac{280.}{\left(3. + \frac{1}{t}\right)^8} - \frac{1680.}{\left(3. + \frac{1}{t}\right)^9} - \frac{6.24054}{4. + \frac{1}{t}} + \frac{8.04565}{\left(4. + \frac{1}{t}\right)^2} - \frac{15.3486}{\left(4. + \frac{1}{t}\right)^3} + \frac{17.8242}{\left(4. + \frac{1}{t}\right)^4} \\ & - \frac{23.7656}{\left(4. + \frac{1}{t}\right)^5} + \frac{1.59265}{5. + \frac{1}{t}} - \frac{2.34847}{\left(5. + \frac{1}{t}\right)^2} + \frac{3.90578}{\left(5. + \frac{1}{t}\right)^3} - \frac{5.02667}{\left(5. + \frac{1}{t}\right)^4} + \frac{5.2}{\left(5. + \frac{1}{t}\right)^5} + \frac{1.31535}{6. + \frac{1}{t}} + \frac{0.173804}{\left(6. + \frac{1}{t}\right)^2} \\ & - \frac{0.280093}{\left(6. + \frac{1}{t}\right)^3} + \frac{0.361111}{\left(6. + \frac{1}{t}\right)^4} - \frac{0.333333}{\left(6. + \frac{1}{t}\right)^5} - \frac{0.603571}{7. + \frac{1}{t}} + \frac{0.0902778}{8. + \frac{1}{t}} - \frac{0.00470312}{9. + \frac{1}{t}} - 3.7422 \times 10^6 t^{14} \\ & + 7.4844 \times 10^6 t^{13} - 8.1648 \times 10^6 t^{12} + 6.51483 \times 10^6 t^{11} - 4.30668 \times 10^6 t^{10} + 2.53181 \times 10^6 t^9 \\ & - 1.38484 \times 10^6 t^8 + 725942.t^7 - 371637.t^6 + 187912.t^5 - 94437.3t^4 + 47325.9t^3 - 23686.9t^2 \\ & + 11850.8t - \frac{1.625}{\frac{1}{t} + 2} + \frac{1}{3\left(\frac{1}{t} + 3\right)}. \quad (50) \end{aligned}$$

Equation (50) has its  $\left[\frac{3}{3}\right]$  Padé approximation in the form:

$$\left[\frac{3}{3}\right](t) = \frac{4.08696t^3 - 0.594203t^2 + 0.5t}{8.17391t^3 + 6.98551t^2 - 0.188406t + 1.0000000000000000}. \quad (51)$$

As  $t = \frac{1}{s}$ , the Equation (51) can be written in terms of the variable  $s$ :

$$\left[\frac{3}{3}\right]\left(\frac{1}{s}\right) = \frac{0.5s^2 - 0.594203s + 4.08696}{1.s^3 - 0.188406s^2 + 6.98551s + 8.17391}. \quad (52)$$

To finalize the solution, we take the inverse of the Laplace transform and arrive at

$$\begin{aligned} \mathcal{L}^{-1}\left[\left[\frac{3}{3}\right]\left(\frac{1}{s}\right)\right] = & 0.5e^{-x} + e^{-2.79658i x} \left( (-1.86795 \times 10^{-14} - 5.21063 \times 10^{-13}i)e^{0.594203x} \right. \\ & \left. - (1.86795 \times 10^{-14} - 5.21063 \times 10^{-13}i)e^{0.594203x} \right). \quad (53) \end{aligned}$$

Some numerical values for our approximate solution using the VIM and LVIM are shown in Table 1. The obtained results indicates an excellent agreement with the exact solution  $y(x)$ .

Now we calculate the solution of (40) using the improved method obtained by combining Adomian method with the VIM, which we called AVIM.

In (40), the non-linear term is  $f(y) = y^3(x)$ . Therefore, the first few Adomian polynomials for  $f(y)$  are  $A_0 = y_0^3(x)$ ,  $A_1 = 3y_1(x)y_0^2(x)$ ,  $A_2 = 3y_2(x)y_0^2(x) + 3y_0(x)y_1^2(x)$ . By Adomian-Variational Iteration Method [AVIM] and using (45), the correction functional of (40) can be written as

$$y_{n+1}(x) = y_n(x) + \int_0^x \frac{1}{2} \left( e^{2s-2x} - 1 \right) \{ y''(s) + 2y'(s) + y(s) + 8 \sum_{k=0}^n A_k(s) - e^{-s} \} ds. \quad (54)$$

By choosing our initial guess to be  $y_0 = A + Bx$ , the first approximation is given by

$$\begin{aligned}
y_1(x) = & -4A^3x - 2A^3e^{-2x} + 2A^3 - 6A^2Bx^2 + 6A^2Bx + 3A^2Be^{-2x} - 3A^2B - 4AB^2x^3 + 6AB^2x^2 \\
& - 6AB^2x - 3AB^2e^{-2x} + 3AB^2 - \frac{Ax}{2} - \frac{1}{4}Ae^{-2x} + \frac{5A}{4} - B^3x^4 + 2B^3x^3 - 3B^3x^2 + 3B^3x \\
& + \frac{3}{2}B^3e^{-2x} - \frac{3B^3}{2} - \frac{Bx^2}{4} + \frac{Bx}{4} - \frac{3}{8}Be^{-2x} + \frac{3B}{8} + \frac{e^{-3x}}{3} - \frac{e^{-2x}}{2} + \frac{1}{6}. \quad (55)
\end{aligned}$$

We use (41) to obtain specific values  $A = -0.5$  and  $B = -0.5$ . In the same manner, using (55), we calculate the next iteration as

$$\begin{aligned}
y_2(x) = & -0.0535714x^7 + 0.6875x^6 - 4.25x^5 + 16.4688x^4 - 1.625e^{-2x}x^3 - 43.0625x^3 - 0.666667e^{-3x}x^2 \\
& + 2.4375e^{-2x}x^2 + 75.5313x^2 - 0.444444e^{-3x}x - 3.25e^{-2x}x - 82.3021x + \frac{e^{-3x}}{3} \\
& - 0.925926e^{-3x} - 1.625e^{-2x} - 40.2344e^{-2x} + 42.952. \quad (56)
\end{aligned}$$

Therefore, we calculate the rest of the iterations until we reach the fifth one. Because of the calculation's length we will not record these iterations. The calculations in Table 2 were based on our fifth iteration. Table 1 shows that the solution by LVIM gives an excellent agreement with the exact solution  $y(x)$  more than the solution by AVIM.

**Table 1.** Numerical results for Example 1 by LVIM.

$x_i$	$y(x_i)$	LVIM	VIM	$E_{VIM}$	$E_{LVIM}$
0	0.5	0.5	0.5	$1.81899 \times 10^{-12}$	$1.81899 \times 10^{-12}$
0.2	0.409365	0.409365	0.409364	$1.28746 \times 10^{-6}$	$2.52365 \times 10^{-12}$
0.4	0.33516	0.33516	0.33511	0.0000497178	$3.06966 \times 10^{-12}$
0.6	0.274406	0.274406	0.274056	0.000350104	$3.26966 \times 10^{-12}$
0.8	0.224664	0.224664	0.223415	0.00124996	$2.97354 \times 10^{-12}$
1	0.18394	0.18394	0.180819	0.00312102	$2.1437 \times 10^{-12}$

**Table 2.** Numerical results for Example 1 by AVIM.

$x_i$	$y(x_i)$	LVIM	AVIM	$E_{AVIM}$	$E_{LVIM}$
0	0.5	0.5	0.5	$7.10543 \times 10^{-15}$	$1.81899 \times 10^{-12}$
0.2	0.409365	0.409365	0.409362	$4.29152 \times 10^{-7}$	$2.52365 \times 10^{-12}$
0.4	0.33516	0.33516	0.335152	$1.22625 \times 10^{-6}$	$3.06966 \times 10^{-12}$
0.6	0.274406	0.274406	0.274412	$7.66562 \times 10^{-5}$	$3.26966 \times 10^{-12}$
0.8	0.224664	0.224664	0.224658	$6.54347 \times 10^{-4}$	$2.97354 \times 10^{-12}$
1.0	0.18394	0.18394	0.184125	$1.89237 \times 10^{-2}$	$2.14370 \times 10^{-12}$

The notation  $E_{method}$  denotes the absolute error using that method. Referring to the calculations in the three tables, we can say that the LVIM is much better than other methods. We also note that the improved AVIM has given significant solutions away from the original. However, the method of Sinc–Galerkin gave good results over the whole period  $[0, 1]$ .

### Example 2. Titchmarch Equation

We consider the Titchmarch model

$$\frac{d^2y(x)}{dx^2} + (\lambda - x^{2m})y(x) = 0, \quad y(0) = y(1) = 0, \quad (57)$$

where  $m$  is a nonnegative integer. For the Sinc solution, we follow the procedure outlines in Section 2 and find numerical solution for (57). The first eigenvalues, listed in Table 3, are computed from the matrix system (20) for

$N = 40$  and three different values of the parameter  $m$ . For the solution using the modified variational iteration method (LVIM), we follow the same steps as in the previous example. Solving  $\lambda$ , we obtain an estimation for the first eigenvalue according to different values of the parameter  $m$ . These results are shown in Table 4. Similar results can also be found in [9].

**Table 3.** Numerical results for Example 1 by Sinc–Galerkin.

$x_i$	$y(x_i)$	Sinc–Galerkin	$E_{Sinc}$
0	0.5	0.5	$3.22370 \times 10^{-8}$
0.2	0.409365	0.409365	$3.11190 \times 10^{-7}$
0.4	0.335160	0.335160	$4.89625 \times 10^{-7}$
0.6	0.274406	0.274405	$2.77628 \times 10^{-6}$
0.8	0.224664	0.224665	$8.55219 \times 10^{-7}$
1.0	0.183940	0.183941	$3.23908 \times 10^{-7}$

The first eigenvalue of the comparison equation  $y''(x) + \lambda y(x) = 0$ ,  $y(0) = 0, y(1) = 0$  is  $\pi^2$ . Table 3 predicts an estimate for the least eigenvalue  $\lambda_1$  that satisfies  $\pi^2 < \lambda_1 < 11$ , which is consistent with the results obtained in [9]. The corresponding eigenfunction to the first eigenvalue  $\lambda_1$  for Equation (57) when  $m = 2$  were computed at some points in its domain, the results are listed in Table 5.

**Table 4.** An estimate to the first eigenvalue  $\lambda_1$  of Equation (57).

$m$	Sinc–Galerkin ( $N = 40$ )	LVIM ( $n = 3$ )
0	10.1363	10.1366
1	10.2754	10.3788
2	10.5655	10.5935

**Table 5.** Comparison of solutions corresponding to the first eigenvalue  $\lambda_1$  of Equation (57).

$x$	Sinc–Solution with ( $N = 40$ )	LVIM ( $n = 3$ )
0	0	0
0.125	0.5461	0.5464
0.250	1.002	1.004
0.375	1.321	1.324
0.500	1.422	1.426
0.625	1.321	1.323
0.750	1.002	1.003
0.875	0.5301	0.5300
1.000	0	0

**Author Contributions:** All authors contributed equally in this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** We wish to thank Bozidar Ivankovic for critically reading the manuscript who provided insight and expertise that were very useful in revising the paper. Also, our appreciation goes to our university (JUST) that supported the second author to complete the master’s thesis, as some results were taken from the master’s thesis for the second author [25].

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chanane, B. Computing the eigenvalues of singular Sturm-Liouville problems using the regularized sampling method. *Appl. Math. Comput.* **2007**, *184*, 972–978. [\[CrossRef\]](#)
2. Bender, C.M.; Orszag, S.A. *Advanced Mathematical Methods for Scientific and Engineers*; McGraw-Hill International Editions; McGraw-Hill: New York, NY, USA, 1987.
3. Celik, I. Approximate calculation of eigenvalues with the method of weighted residuals-collocation method. *Appl. Math. Comput.* **2005**, *160*, 401–410.
4. Guseinov, G.S.; Karaca, I.Y. Instability intervals of a Hill's equation with piecewise constant and alternating coefficient. *Comput. Math. Appl.* **2004**, *47*, 319–326. [\[CrossRef\]](#)
5. El-Gamel, M.; Zayed, A.I. Sinc-Galerkin method for solving nonlinear boundary-value problems. *Comput. Math. Appl.* **2004**, *48*, 1285–1298. [\[CrossRef\]](#)
6. Saadatmandi, A.; Razzaghi, M.; Dehghan, M. Sinc-Galerkin solution for nonlinear two-point boundary value problems with applications to chemical reactor theory. *Math. Comput. Model.* **2005**, *42*, 1237–1244. [\[CrossRef\]](#)
7. Stenger F. A Sinc-Galerkin method of solution of boundary value problems. *Math. Comput.* **1979**, *33*, 85–109.
8. Mohsen, A.; El-Gamel, M. On the Galerkin and collocation methods for two-point boundary value problems using sinc bases. *Comput. Math. Appl.* **2008**, in press. [\[CrossRef\]](#)
9. Alquran, M.; Al-Khaled, K. Approximations of Sturm-Liouville eigenvalues using sinc-Galerkin and differential transform methods. *Appl. Appl. Math.* **2010**, *5*, 128–147.
10. He, J.H. Variational iteration method—A kind of non-linear analytical technique: Some examples. *Int. J. Non-Linear Mech.* **1999**, *34*, 699–708. [\[CrossRef\]](#)
11. He, J.H. Variational iteration method?some recent results and new interpretations. *J. Comput. Appl. Math.* **2007**, *207*, 3–17. [\[CrossRef\]](#)
12. He, J.H.; Wu, X.H. Variational iteration method: New development and applications. *Comput. Math. Appl.* **2007**, *54*, 881–894. [\[CrossRef\]](#)
13. Abassy, T.A.; El-Tawil, M.A.; El Zoheiry, H. Toward a modified variational iteration method. *J. Comput. Appl. Math.* **2007**, *207*, 137–147. [\[CrossRef\]](#)
14. Jin, L. Application of modified variational iteration method to the Bratu-type problems. *Int. J. Contemp Math. Sci.* **2010**, *5*, 153–158.
15. Islam, S.U.; Haq, S.; Ali, J. Numerical solution of special 12th-order boundary value problems using differential transform method. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, in press. [\[CrossRef\]](#)
16. Abbasbandy, S. A new application of He's variational iteration method for quadratic Riccati differential equation by using Adomian's polynomials. *J. Comput. Appl. Math.* **2007**, *207*, 59–63. [\[CrossRef\]](#)
17. Wazwaz, A.M. The variational iteration method for solving two forms of Blasius equation on a half-infinite domain. *Appl. Math. Comput.* **2007**, *188*, 485–491. [\[CrossRef\]](#)
18. Mukhtarov, O.S.; Yücel, M. A Study of the Eigenfunctions of the Singular Sturm-Liouville Problem Using the Analytical Method and the Decomposition Technique. *Mathematics* **2020**, *8*, 415. [\[CrossRef\]](#)
19. Qadir, R.R.; Jwamer, K.H.F. Refinement Asymptotic Formulas of Eigenvalues and Eigenfunctions of a Fourth Order Linear Differential Operator with Transmission Condition and Discontinuous Weight Function. *Symmetry* **2019**, *11*, 1060. [\[CrossRef\]](#)
20. Khashshah, M.M.; Syam, M.I.; Al Mokhmari, A. A Reliable Method for Solving Fractional Sturm-Liouville Problems. *Mathematics* **2018**, *6*, 176. [\[CrossRef\]](#)
21. Alsaedi, A.; Alsulami, M.; Srivastava, H.M.; Ahmad, B.; Ntouyas, S.K. Existence Theory for Nonlinear Third-Order Ordinary Differential Equations with Nonlocal Multi-Point and Multi-Strip Boundary Conditions. *Symmetry* **2019**, *11*, 281. [\[CrossRef\]](#)
22. Stenger, F. *Numerical Methods Based on Sinc and Analytic Functions*; Springer: New York, NY, USA, 1993.
23. Eggert, N.; Jarratt, M.; Lund, J. Sinc function computation of the eigenvalues of Sturm-Liouville problems. *J. Comput. Phys.* **1987**, *69*, 209–229. [\[CrossRef\]](#)
24. McArthur, K.M. A Collocative Variation of the Sinc-Galerkin Method for Second Order Boundary Value Problems. In *Computation and Control*; Progress in Systems and Control Theory; Birkhäuser: Boston, MA, USA, 1989; Volume 1.
25. Hazaimah, A. Solution of Sturm-Liouville Differential Equation via the Use of Variational Iteration Method. Master's Thesis, Jordan University of Science and Technology, Ar-Ramtha, Jordan, May 2020.



26. Adomian, G. A review of the decomposition method and some recent results for nonlinear equations. *Math. Comput. Model.* **1990**, *13*, 17–43. [[CrossRef](#)]
27. Singh, N.; Kumar, M. Adomian decomposition method for computing eigen-values of singular Sturm-Liouville problems. *Natl. Acad. Sci. Lett.* **2013**, *36*, 311–318. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).