On the numerical solution of the Lane-Emden, Bratu and Troesch equations

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF KWAZULU-NATAL
FOR THE DEGREE OF MASTER OF SCIENCE
IN THE COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

by

Phumla Remember Sithole

School of Mathematics, Statistics and Computer Science

December 2020

Declaration

This work was carried out in the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg under the supervision of Prof. P. Sibanda and Dr. S. P. Goqo from February 2016 to December 2020.

I hereby declare that, except where due credit is given, no portion of this work has been presented wholly or in part for the award of any degree or qualification at University of KwaZulu-Natal or any other institution.

Phumla R. Sithole Date 28/07/2021

[redacted]

Prof. P. Sibanda Date 3-November-2021

[redacted]

Dr. S. P. Goqo Date

Acknowledgements

Firstly, I would like to thank the Almighty God for strength to achieve my goals. I would like to thank my supervisors Prof. P. Sibanda and Dr. S.P. Goqo for their amazing support, guidance during the course of my study, for being patient with me and never giving up on me.

I would like to thank my brother, Thobani and sister Zandile for being the pillar of strength during difficult times and for always seeing the best in me. I would like to thank my aunt for helping me with my child for me to finish my Masters work. I would like to thank amazing women in my life, Namhla Sihlali, Xoliswa Mkhwanazi and many more for great support.

Abstract

Many engineering and physics problems are modelled using differential equations, which may be highly nonlinear and difficult to solve analytically. Numerical techniques are often used to obtain approximate solutions. In this study, we consider the solution of three nonlinear ordinary differential equations; namely, the initial value Lane-Emden equation, the boundary value Bratu equation, and the boundary value Troesch problem. For the Lane-Emden equation, a comparison is made between the accuracy of solutions using the finite difference method and the multi-domain spectral quasilinearization method along with the exact solution. We found that the multi-domain spectral quasilinearization method gave a better solution. For the Bratu problem, a comparison is made between the spectral quasilinearization method and the higher-order spectral quasilinearization method. The higher-order spectral quasilinearization method gave more accurate results. The Troesch problem is solved using the higher-order spectral quasilinearization method and the finite difference method. The solutions obtained are compared in terms of accuracy. Overall, the higher-order spectral quasilinearization method and multi-domain spectral quasilinearization method gave the accurate solutions, making these two methods to be the most reliable for these three problems.

Contents

List of Figures

List of Tables

List of Abbreviations

ADM	Adomian's decomposition method
FDM	Finite difference methods
HBM	Harmonic balance method
HO-SQLM	Higher-order spectral quasilinearization method
HPM	Homotopy perturbation method
IVP	Initial value problems
MD-SQLM	Multi-domain spectral quasilinearization method
ODE	Ordinary differential Equations
QLM	Quasilinearization method
SQLM	Spectral quasilinearization method
VIM	Variation iteration method

Chapter 1

Introduction

Differential equations are used to model problems that arise in physics and engineering. In many cases, the equations are nonlinear, and approximate solutions are obtained using different numerical techniques. These techniques have improved our ability to analyze complicated mathematical models. These methods include, but are not limited to, the finite difference method, finite volume method, spectral methods, quasilinearization method, variational iteration method, Adomian decomposition method, and homotopy perturbation method.

In this chapter, we give a brief review of some common numerical techniques and some relevant instances of their use in the literature.

We begin by reviewing two of the most commonly used techniques, the finite difference method and the finite element method. Then we discuss in detail the main quasilinearization methods that are used in this study.

1.1 The finite difference method

The finite difference method is one of the popular methods used to solve differential equations. The method dates back to 1930 when it was used to obtain solutions of the Dirichlet biharmonic equations [1]. It is derived from a Taylor series expansion that approximate the derivatives of the unknown functions using forward, backward, and central difference schemes [2]. In the finite difference method, independent variables are discretized in space or in time, with step size Δx or Δt. The finite difference method has three common versions; namely, the implicit finite difference method which uses the backward difference approximation, the explicit finite difference method which uses the forward difference approximation, and the Crank-Nicholson scheme which uses the central difference scheme. Several authors have used the finite difference method to solve mathematical problems. Hickson et al. [3] presented a finite difference scheme for multi-layered materials and found the method to be reliable, accurate, and easy to use in practice. Wong and Li [4]solved the Helmholtz equation using the finite difference method, and showed that the method can be used for high-frequency problems with no requirement of fine step size. They obtained accurate numerical solutions that showed the method to be efficient. Moczo et al. [5] however, showed that the finite difference method has limitations, such as when the boundary conditions are complex. Wang and Lin [6]noted that the implicit scheme is more stable than the explicit scheme, it also produces more accurate solutions although it requires more computational time than the explicit scheme. Cui [7] solved the one-dimensional diffusion equation using the compact finite difference method by means of the Grunwald-Letnikov discretization scheme. They found the compact finite difference method to be highly accurate. Tayakout et al. [8] established a catalytic membrane reactor model and used the finite difference method to obtain the solution of the dynamic system. Yuste and Acedo [9] used an explicit finite difference method to solve the fractional diffusion equation. They used the forward Euler difference scheme with a Fourier- Von Neumann technique for the conditions of stability. Chen et al. [10] used the finite difference method to solve the fractional reaction-subdiffusion equation. They presented both implicit and explicit difference schemes. They also investigated the stability and the convergence of the schemes using Fourier analysis. They found that the method is accurate. Zhuang et al. [11] solved the anomalous sub-diffusion equation using the finite difference scheme for an implicit numerical method. They noted that the method was convergent and unconditionally stable. Martino et al. [12] applied the finite difference method to the Fisher equation using the Crank-Nicholson approximation to obtain the solution.

1.2 The finite element method

The finite element method has its origin in the work of Euler in the sixteenth century, although Courant [13] was the first to give the modern formulation of the method in 1943. The finite element method was further developed independently to solve problems in civil engineering and aerospace [14]. The finite element method consists of two numerical techniques; namely, the Galerkin and Ritz methods. In the Galerkin method, the variables are approximated inside the element using continuous piecewise functions. In the Ritz method, calculus of variation is used to transform the problem into integral form [17]. The finite element method splits the domain of the problem into n finite elements, which are connected at the nodes. It is among the most commonly used methods for solving engineering and mathematical problems, as it can be used for problems with domains that are geometrically complicated, for those with complex boundary conditions, and for coupled nonlinear problems [15]. The finite element method increases the number of unknown terms inside a cell to give a higher-order solution [18]. Alrabeei and Musthafa [19] used the finite element method to solve the two and three-dimensional linear elastostatic model in order to estimate the elastic mechanical response. Hiltunen [20] presented a solution for particulate two-phase flow and showed that the method is reliable and accurate. Goldstein [21] solved the Helmholtz equation using the finite element method and found the method to be versatile and reliable. Marchardise and Remacle [22] used an implicit stabilized finite element method to obtain a solution for an incompressible two-phase flow problem. They found that the method was flexible, simple, and accurate. Nagrath et al. [23] solved the equation for incompressible bubble dynamics using the Galerkin finite element method for the discretization of the governing equations. They noted that the method was robust. Tezduyar [24] used the finite element method to solve some flow problems with boundaries and an interface that moves and found that the method was accurate and efficient. Boncut [25] solved the fluid flow equations using the Ritz finite difference method. Chowdhury and Narasimhan [26] also used the Ritz finite difference method and presented a solution for the processes of fracture and delamination in solids. They found that the method was effective and showed a degree of robustness.

1.3 The quasilinearization method

The quasilinearization technique, introduced in 1965 by Bellman and Kalaba [27], is useful for finding approximate solutions of a nonlinear differential equation[28]. It was introduced as a general statement of the Newton-Raphson method [29]. The quasilinearization method (QLM) has been used extensively. For example, Koleva and Vulkov [30] proposed a quasilinearization scheme that was coupled with Rothe's method for solving nonlinear parabolic equations. The rate of convergence and numerical errors were tested on problems with closed-form solutions. Their scheme is simple to apply and the results showed that, with a good initial guess, it was possible to obtain quadratic convergence [30]. Wang et al. [31] applied the quasilinearization method to second-order impulsive differential equations, while Yakar and Koksal [32] employed it to solve nonlinear problems. Vijesh et al. [33] used a modified quasilinearization method to solve fractional differential equations. They found that the method was simple to apply and was reliable and accurate. To demonstrate the use of the quasilinearization method (QLM), consider the nonlinear differential equation

$$G[f(\eta), f'(\eta), f''(\eta), \cdots, f^{(n)}(\eta)] = 0, \qquad \eta \in (a, b) \tag{1.1}$$

with boundary conditions

$$\begin{aligned} H_k[f(a), f'(a), f''(a), \cdots, f^{(n-1)}(a)] &= 0, \qquad k = 1, 2, 3, \cdots, m, \\ H_k[f(b), f'(b), f''(b), \cdots, f^{(n-1)}(b)] &= 0, \qquad k = m+1, m+2, \cdots, n, \end{aligned}$$

where G is a nonlinear ordinary differential operator of $f(\eta)$ and the n^{th} derivatives, $H_{a,k}$, $H_{b,k}$ are nonlinear functions of $f(\eta)$ and its $n-1$ derivatives at $\eta = a$ and $\eta = b$ for $l = 1, 2, \cdots, n-1$. The QLM approach assumes that the difference between the approximation of the solution at the current iteration (denoted by $f_r(\eta)$) and the previous iterations (denoted by $f_{r+1}(\eta)$) is small. The difference between the derivatives at the subsequent iterations levels, $f_{r+1}^{(p)} - f_r^{(p)}$, is also assumed to be small. Expanding equation

(1.1) using the Taylor series gives

$$G[f_r, f_r', \cdots, f_r^{(n)}] \approx G[f_r, f_r', \cdots, f_r^{(n)}]$$
$$+(f_{r+1} - f_r, f_{r+1}' - f_r', \cdots, f_{r+1}^{(n)} - f_r^{(n)})\nabla G[f_r, f_r', \cdots, f_r^{(n)}]$$

where ∇ is a vector of partial derivatives defined by

$$\nabla = \left\{\frac{\partial}{\partial f}, \frac{\partial}{\partial f'}, \frac{\partial}{\partial f''}, ..., \frac{\partial}{\partial f^{(n)}}\right\}.$$

The equation can be written in the compact form

$$G[f_r(\eta), f_r'(\eta), \cdots, f_r^{(n)}(\eta)] + \sum_{p=0}^{n} \frac{\partial G}{\partial f^{(p)}}[f_r(\eta), f_r'(\eta), \cdots, f_r^{(n)}(\eta)](f_{r+1}^{(p)}(\eta) - f_r^{(p)}) = 0.$$

The QLM determines the approximate solutions at the $(r+1)$ iteration level as the solution of the linear equation

$$G[f_r(\eta), f_r'(\eta), \cdots, f^{(n)}(\eta)] + \sum_{p=0}^{n} \frac{\partial G}{\partial f^{(p)}}[f_r(\eta), f_r'(\eta), \cdots, f_r^{(n)}(\eta)](f_{r+1}^{(p)}(\eta) - f_r^{(p)}(\eta)) = 0. \tag{1.2}$$

We linearize the boundary conditions

$$H_{a,k}[f_r(a), f_r'(a), \cdots, f_r^{(n)}(a)] + \sum_{p=0}^{n-1} \frac{\partial H_{a,k}}{\partial f^{(p)}}[f_r(a), f_r'(a), \cdots, f_r^{(n)}(a)](f_{r+1}^{(p)}(a) - f_r^{(p)}(a)) = 0,$$

for $k = 1, 2, \cdots, m$ and

$$H_{b,k}[f_r(b), f_r'(b), \cdots, f_r^{(n)}(b)] + \sum_{p=0}^{n-1} \frac{\partial H_{b,k}}{\partial f^{(p)}}[f_r(b), f_r'(b), \cdots, f_r^{(n)}(b)](f_{r+1}^{(p)}(b) - f_r^{(p)}(b)) = 0, \tag{1.3}$$

for $k = 1, 2, \cdots, m$.

It is convenient to write equations (1.2) and (1.3) in the form

$$a_{n,r}(\eta)f_{r+1}^{(n)}(\eta) + a_{n-1,r}(\eta)f_{r+1}^{(n-1)}(\eta) + \cdots + a_{1,r}(\eta)f_{r+1}'(\eta) + a_{0,r}(\eta)f_{r+1}(\eta) = R_r(\eta),$$

$$\alpha_{n-1,k}f_{r+1}^{(n-1)}(a) + \alpha_{n-2,k}f_{r+1}^{(n-2)}(a) + \cdots + \alpha_{1,k}f_{r+1}'(a) + \alpha_{0,k}f_{r+1}(a) = R_{a,k},$$

$$\beta_{n-1,k}f_{r+1}^{(n-1)}(b)+\beta_{n-2,k}f_{r+1}^{(n-2)}(b)+\cdots+\beta_{1,k}f'_{r+1}(b)+\beta_{0,k}f_{r+1}(b)=R_{b,k},$$

where

$$a_{p,r}(\eta)=\frac{\partial G}{\partial f^{(p)}}[f_r(\eta),f'_r(\eta),f''_r(\eta),\cdots,f_r^{(n)}(\eta)],$$

$p=0,1,2,\cdots,n,$

$$R_r(\eta)=\sum_{p=0}^{n}\frac{\partial G}{\partial f^{(p)}}[f_r(\eta),f'_r(\eta),f''_r(\eta),\cdots,f_r^{(n)}(\eta)]-G[f_r(\eta),f'_r(\eta),f''_r(\eta),\cdots,f_r^{(n)}(\eta)],$$

$$\alpha_{p,k}=\frac{\partial H_{a,k}}{\partial f^{(p)}}[f_r(a),f'_r(a),f''_r(a),\cdots,f_r^{(n)}(a)],$$

$p=0,1,2,\cdots,n-1,\;\; k=1,2,...,m,$

$$\beta_{p,k}=\frac{\partial H_{b,k}}{\partial f^{(p)}}[f_r(b),f'_r(b),f''_r(b),\cdots,f_r^{(n)}(b)],$$

$p=0,1,2,\cdots,n-1,\;\; k=m+1,m+2,\cdots,n$

$$R_{a,k}=\sum_{p=0}^{n-1}\frac{\partial H_{a,k}}{\partial f^{(p)}}[f_r(a),f'_r(a),f''_r(a),\cdots,f_r^{(n-1)}(a)]-H_{a,k}[f_r(a),f'_r(a),f''_r(a),\cdots,f_r^{(n-1)}(a)],$$

$$R_{b,k}=\sum_{p=0}^{n-1}\frac{\partial H_{b,k}}{\partial f^{(p)}}[f_r(b),f'_r(b),f''_r(b),\cdots,f_r^{(n-1)}(b)]-H_{b,k}[f_r(b),f'_r(b),f''_r(b),\cdots,f_r^{(n-1)}(b)]$$

for $k=m+1,m+2,\cdots,n.$

1.4 Spectral methods

Spectral methods involve approximating unknown functions by truncated orthogonal functions [34]. They differ from the finite difference and finite element methods, which both use local basis functions, in that spectral methods use global basis functions to represent the outputs of differential equations. Spectral methods are an improvement in the class of discretization schemes called the method of weighted residuals. The methods give solutions with a high-resolution [35]. The trial functions used in spectral methods may be trigonometric, monomials, or Chebyshev or Legendre polynomials. Spectral methods include the tau, Galerkin, and collocation approaches, which are distinguished by the way the test functions are chosen. In the Galerkin approach, the trial and the test functions are the same, while in the collocation approach the test functions are Dirac delta functions, and although the tau approach is similar to the Garlekin approach, it treats the boundary conditions differently. The collocation approach requires the differential equations to be satisfied at collocation points, while the Galerkin approach is similar to a least-squares approximation. The tau method was developed as a modification of the Galerkin method for problems with boundary conditions that are non-periodic. Although the tau method is said to be the most complicated, it is useful for solving constant coefficient problems [36]. Gottlieb and Gottlieb [37] applied the spectral method to solve equations for compressible reactive flows. They found that the it gives stable solutions. Pohl, Wenzel and Karrenbach [38] presented a solution for two-dimensional generalized coordinates, wherein they applied the Chebyshev pseudospectral method to the wave equations. They noted that the stability of this method relies on minimum node spacing. Doha et al. [39] solved multi-term fractional orders differential equations using the Chebyshev spectral method, and presented results that indicated the method's good accuracy. Gardner et al. [40] used a modified tau spectral method to solve eigenvalue problems. They found that the method converges rapidly. Ghoreishi and Yazdani [41] provided a numerical solution for fractional differential equations, which utilized the interpolating expansion to approximate the integral term of the equations. They found that the method is efficient and reliable. Bhrawy and Alghamdi [42] used the tau spectral method to solve the time-fractional heat equation with nonlocal conditions. They found the method to be accurate and effective. Shen [43] presented a solution for second and fourth-order elliptic equations using the spectral Galerkin method with Legendre polynomials and found that the method is accurate and reliable.

In the spectral technique, an approximating series is obtained with the original differential equation being satisfied precisely at different collocation nodes. There is a link between the choice of the basis functions and the positions of the points in the domain.

The function $F(x)$ can be represented as a series expansion

$$F_n(x) = \sum_{m=0}^{n} \hat{F}_m G_m(x), \tag{1.4}$$

where $G_m(x)$ are the basis functions, n is the total number of collocation points, and $\hat{F}_m$ are the spectral coefficients. The most commonly used collocation points are Chebyshev Gauss-Lobatto collocation points, given by

$$x_i = \cos\left(\frac{\pi i}{N}\right), \qquad i = 0, 1, \cdots, N. \tag{1.5}$$

The last decade has seen concerted efforts to construct new techniques incorporating quasilinearization and solving the resulting equations using spectral methods. The new techniques that are of interest in this study include the spectral quasilinearization method, and the higher-order and multi-domain spectral quasilinearization methods. A brief discussion of each technique is given below.

1.4.1 The spectral quasilinearization method

The spectral quasilinearization method (SQLM) is a combination of the quasilinearization method and Chebyshev spectral collocation method [44]. This technique was introduced by Motsa and Sibanda in 2013 to solve nonlinear boundary value problems [45]. In the method, the nonlinear terms are linearized using the Taylor series expansion to obtain a system of linear equations. The Chebyshev pseudospectral method is then used to solve the linearized equations. Several researchers have used the spectral quasilinearization method in their studies. RamReddy et al. [46] solved the equations that describe the mixed convection flow of a micropolar fluid over a permeable vertical plate using the spectral quasilinearization method. Alharbey et al. [47] solved the flow equations in a non-Darcy porous medium with convective boundary conditions. They monitored the convergence of the solution by calculating the residual errors of the partial differential equations. They found that the spectral quasilinearization method is accurate and efficient. Mondal and Sibanda [48] used the spectral quasilinearization method to solve nonlinear transport equations.

These equations arise in the study of entropy generation where the Cattaneo-Christov heat flux model is used. They also presented an analysis of the convergence of the method, whereby they could show that the method converges rapidly and is accurate. Motsa et al. [49] used both the spectral quasilinearization and spectral relaxation methods separately to investigate the unsteady heat transfer in a nanofluid over a permeable stretching or shrinking surface. They observed that the spectral quasilinearization method performed better than did the spectral relaxation method.

1.4.2 The multi-domain spectral quasilinearization method

The multi-domain spectral quasilinearization method is a linearization-based technique. The resulting linear equations are integrated into numerous sub-intervals using the Chebyshev spectral collocation method. The basis functions are the Lagrange interpolation polynomials [50]. Magagula et al. [51] used the multi-domain bivariate pseudospectral method to solve nonlinear differential equations using Legendre-Gauss-Lobatto grid points in space and time. They found the multi-domain spectral quasilinearization method to be an accurate method, by comparing it with the solutions obtained by the bivariate spectral quasilinearization method. They also noted that the method is efficient. Goqo et al. [52] applied the multi-domain spectral quasilinearization method to the equations of laminar natural convective flow from a vertical flat plate. They found the method to be efficient and accurate. Ayano et al. [53] studied the problem of an electrically conducting and steady incompressible micropolar fluid. They showed that the multi-domain spectral quasilinearization method can be used to solve such fluid flow problems; the method being accurate and efficient. Oyelakin et al. [54] were the first to use the multi-domain bivariate spectral quasilinearization method to investigate non-Darcian mixed convection flow, heat, and mass transfer in a non-Newtonian fluid. Their interest was in determining skin friction, fluid properties, and heat and mass coefficients. The solution they obtained showed that the method converges rapidly, produces accurate results, and is simple to use. Oyelakin et al. [55] also used the same multi-domain spectral quasilinearization method to solve a system of partial differential equations when they investigated mixed convection in a Casson nanofluid with heat generation.

1.4.3 The higher-order spectral quasilinearization method

The higher-order spectral quasilinearization method is a modification of the spectral quasilinearization method. In this case, a higher-order Newton-like iteration approach is used to solve the nonlinear problems [56]. In the higher-order spectral quasilinearization method, the equation is separated into its linear and nonlinear parts. The Taylor method is then applied to the nonlinear part. The spectral collocation is imposed on the iterative scheme with Chebyshev-Gauss-Lobato nodes. Motsa et al. [57] used the higher-order spectral quasilinearization method to solve the nonlinear model of catalytic reactions. They found that the method converges rapidly and gave accurate results. The higher-order approach has been used by other researchers. Chun [58] applied the approach with the Adomian decomposition method to solve nonlinear equations. Motsa and Sibanda [56] used the higher-order scheme with the homotopy analysis method to obtain the solutions for a highly nonlinear problem.

We present the higher-order quasilinearization method for a k-th order ordinary differential equation. A nonlinear differential equation is obtained with a solution of $y(\eta)$

$$L\left[y^{(k)}\right] + N\left[y^{(k)}\right] = 0, \qquad k = 0, 1, \cdots, n, \qquad \eta \in [a, b], \tag{1.6}$$

where L is a linear component and N is non-linear component of the ordinary differential equations. The Taylor's series expansion is applied to the non-linear component $N\left[y^{(k)}\right]$ about the point y_i, where y_i is an initial guess of the solution for the ordinary differential equation. This gives,

$$N\left[y^{(k)}\right] \approx N\left[y_i^{(k)}\right] + \sum_{k=0}^{n} \alpha_{k,i}\left(y^{(k)} - y_i^{(k)}\right) + H\left[y^{(k)}, y_i^{(k)}\right],$$

where

$$\alpha_{k,i} = \frac{\partial N(y_i)}{\partial y^{(k)}} \qquad \text{and}$$
$$H\left[y^{(k)}, y_i^{(k)}\right] = N\left[y^{(k)}\right] - N\left[y_i^{(k)}\right] - \sum_{k=0}^{n} \alpha_{k,i}\left(y^{(k)} - y_i^{(k)}\right).$$

Equation (1.6) becomes

$$L\left[y^{(k)}\right]+N\left[y_i^{(k)}\right]+\sum_{k=0}^{n}\alpha_{k,i}\left(y^{(k)}-y_i^{(k)}\right)+H\left[y^{(k)},y_i^{(k)}\right]=0.$$

The above equation can be written as

$$L\left[y^{(k)}\right]+\sum_{k=0}^{n}\alpha_{k,i}y^{(k)}+H\left[y^{(k)},y_i^{(k)}\right]=\phi\left[y_i^{(k)}\right], \tag{1.7}$$

where

$$\phi\left[y_i^{(k)}\right]=-N\left[y_i^{(k)}\right]+\sum_{k=0}^{n}\alpha_{k,i}y_i^{(k)}.$$

We use the quasi-linearization method of Bellman and Kalaba[27] to approximate equation (1.7), so we obtain

$$L\left[y_{r+1}^{(k)}\right]+\sum_{k=0}^{n}\alpha_{k,i}y_{r+1}^{(k)}+H\left[y_r^{(k)},y_i^{(k)}\right]+\sum_{k=0}^{n}\left(y_{r+1}^{(k)}-y_r^{(k)}\right)\frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)},y_i^{(k)}\right]=\phi\left[y_i^{(k)}\right].$$

When the terms of the known solution are transposed to the right, we get

$$\begin{aligned}L\left[y_{r+1}^{(k)}\right]+\sum_{k=0}^{n}\alpha_{k,i}y_{r+1}^{(k)}+\sum_{k=0}^{n}y_{r+1}^{(k)}\frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)},y_i^{(k)}\right]&=\phi\left[y_i^{(k)}\right]-H\left[y_r^{(k)},y_i^{(k)}\right]\\&\quad+\sum_{k=0}^{n}y_r^{(k)}\frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)},y_i^{(k)}\right].\end{aligned} \tag{1.8}$$

The Chebyshev spectral collocation method is used to solve equation (1.8). The method uses the spectral differentiation matrix, which is defined at the Chebyshev-Gauss-Lobatto nodes as

$$t_j=\cos\frac{\pi j}{N_t}\bigg|_{j=0}^{N_t},$$

with N_t being the number of collocation points. The linear transformation below is used to convert the region $\eta \in [a,b]$ to $t \in [-1,1]$

$$\eta = \frac{1}{2}(b-a)t + \frac{1}{2}(b+a).$$

The spectral collocation is applied to equation (1.8), with the derivatives evaluated at the Chebyshev-Gauss-Lobatto nodes as

$$\frac{d^{(k)}y}{d\eta^{(k)}} = \sum_{p=0}^{N} D_{jp}^{(k)} y_j(t_p), \qquad k = 0, 1, \cdots, n,$$

where $D_{jp}^{(k)}$ is the k-th order Chebyshev differential matrix in t of size $(N+1)\times(N+1)$. This collocation results in

$$\begin{aligned} L\left[D^{(k)}y_{r+1}\right] + \sum_{k=0}^{n} \alpha_{k,i} D^{(k)} y_{r+1} + \sum_{k=0}^{n} D^{(k)} y_{r+1} \frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)}, y_i^{(k)}\right] &= \phi\left[y_i^{(k)}\right] \\ -H\left[y_r^{(k)}, y_i^{(k)}\right] + \sum_{k=0}^{n} y_r^{(k)} \frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)}, y_i^{(k)}\right]&. \end{aligned}$$

In matrix form the equation can be written as

$$\mathbf{M}_r Y_{r+1} = \Phi_r,$$

where

$$\begin{aligned} \mathbf{M}_r &= L\left[D^{(k)}\right] + \sum_{k=0}^{n} \alpha_{k,i} D^{(k)} + \sum_{k=0}^{n} D^{(k)} \frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)}, y_i^{(k)}\right] \\ Y_{r+1} &= [y_0, y_1, \cdots, y_N]^T \\ \Phi_r &= \phi\left[y_i^{(k)}\right] - H\left[y_r^{(k)}, y_i^{(k)}\right] + \sum_{k=0}^{n} y_r^{(k)} \frac{\partial H}{\partial y_i^{(k)}}\left[y_r^{(k)}, y_i^{(k)}\right]. \end{aligned}$$

1.5 The Adomian decomposition method

In 1980 Adomian introduced a decomposition method to solve linear and nonlinear ordinary differential equations. This method is known as the Adomian decomposition method

[59]. The Adomian decomposition method can be used to compute analytic solutions for an extensive class of nonlinear differential equations with no need for perturbation, linearization or discretization [60]. The Adomian decomposition method gives solutions in terms of a rapidly converging power series [61]. In the Adomian decomposition method (ADM), the equation is first split into linear and nonlinear parts. The highest-order derivative operator that is contained in the linear operator is inverted. The nonlinear functions are decomposed in terms of Adomian polynomials and the successive terms of the series solution are determined. Jafari and Daftardar-Gejji [62] used the Adomian decomposition method to solve linear and nonlinear fractional diffusional equations. They found the ADM to be a very powerful tool that can handle both linear and nonlinear fractional partial differential equations well. They identified a limitation of this method being in the computation of the Adomian polynomials, which they overcame by using Mathematica to obtain the polynomials. Ismail et al. [63] solved the Burgers-Huxley and Burgers-Fisher equations using the Adomian decomposition method. They noted that the advantage that method could be used without the need for complex calculations, as operations were largely simple and elementary operations. Dhaigude et al. [64] used the Adomian decomposition method to solve the Benjamin-Bona-Mahony-Burgers equation. They found the method to be accurate, efficient, and reliable. Sánchez Cano [65] used the Adomian decomposition method to solve a coupled system problem and nonlinear problems. It was noted that the method is a reliable technique for solving nonlinear initial value problems and coupled systems. They also noted that there is no solution in closed form. Evans and Raslan [66] used the Adomian decomposition method to solve the delay differential equations. They showed the method to be reliable and give accurate results. Kaya [67] solved the partial differential using the Adomian decomposition method, and pointed out that the decomposition method does not require perturbation theory, linearization, or weak nonlinear assumptions. The method was also found to converge rapidly and be highly accurate.

1.6 The variational iteration method

The variational iteration method was first introduced in 1999 by He, when it was used to obtain the series solution of a nonlinear differential equation [68]. The variational iteration method is related to the well known Lagrange multiplier method [69] and has been applied successfully to ordinary differential equations [70]. According to He [70], the underlying principle is the application of a correction function to the problem. The correction function is formulated by using a Lagrange multiplier, λ, where λ is selected to give a solution that is better than the initial approximation. Stationary conditions are applied to the correction formula to obtain the optimal Lagrange multiplier for the problem. The variational iteration method has since been used by other authors to solve various physics and engineering problems. For example, Wazwaz [71] was interested in the utility of the method in solving physics and engineering problems, and used the method to solve the linear and nonlinear wave equations. Wazwaz stated that the variational iteration method greatly reduces the number of calculations, while maintaining high solution accuracy. He also found that the variational iteration method converges quickly. Shang [72] obtained approximate solutions for n-th order differential equations using the variational iteration method. The advantage of the variational iteration method is, according to Shang, the initial approximation being a free choice, He used this principle to formulate the initial approximation, excluding the parameters that were not known. He showed that the method is efficient and easy to work with. Abassy et al. [73], however, saw disadvantages in the variational iteration method in the computations being repeated and unnecessary terms being computed.

1.7 Consistency, stability and convergence

The characteristics of a good numerical method are that it should be convergent, consistent, and stable. The finite difference method has been shown in the literature to have these properties. A numerical scheme is consistent if the difference between the approximate solution and the exact solution approaches zero as the step size becomes smaller.

Consider the differential equation

$$f(u) = 0, \tag{1.9}$$

where $u(x)$ is continuous and its differential is consistent with an approximate solution

$$\hat{f}(u) = 0. \tag{1.10}$$

Then

$$\hat{f}(u) - f(u) = 0, \tag{1.11}$$

as the stepsize $\Delta x \to 0$ [75]. The finite difference method is consistent if the local truncation error, τ , satisfies the condition

$$\|\tau_{\Delta x, \Delta t}\| \to 0 \quad as \quad \Delta x, \Delta t \to 0. \tag{1.12}$$

If a scheme can be discretized such that $\|\tau_{\Delta x, \Delta t}\| = \mathcal{O}(\Delta x^m) + \mathcal{O}(\Delta t^n)$ for some $m, n > 0$, then the scheme is consistent.

A numerical technique is stable if the difference between the approximate solution and the exact solution of the equation is very small and does not increase as the number of nodes increases [75]. A finite difference scheme in the form $f_{\Delta_x, \Delta_t}(u_i^n)$ is stable in the region Λ, for all positive time T, with a fixed constant p. The condition for stability is

$$||u^n||_{\Delta} t \leq p||u^i||\Delta x, \tag{1.13}$$

where p is a constant and is independent of the step sizes, $0 \leq n\Delta t \leq T$ for $\Delta x, \Delta t \in \Lambda$. According to Lax's equivalence theorem, when a numerical scheme is consistent and stable, it is guaranteed that scheme is convergent [76]. In the finite difference method, convergence is achieved if the approximations u_i^n converge to the analytical solution $u(x, t)$ as $\Delta x, \Delta t$ approach 0.

1.8 Aims and objectives

The aim of this study is to use numerical methods to solve three highly nonlinear ordinary differential equations; namely, the Bratu, the Lane-Emden, and the Troesch equations. The specific objective is to compare the performance of different numerical schemes when used to solve these nonlinear differential equations. The Lane-Emden equation is solved using the finite difference method and the multi-domain spectral quasilinearization method. The Bratu equation is solved using the spectral quasilinearization method and higher-order quasilinearization method. The Troesch problem is solved using the finite difference method and the higher-order spectral quasilinearization method. The choice of methods used for each problem is informed by the need to determine how the methods perform for the specific problems and relative to each other. The accuracy of each method is further examined by evaluating the residual errors.

The dissertation is organized as follows; In Chapter 2, a brief literature review of the Lane-Emden equation is presented. We use the finite difference method and the multi-domain spectral quasilinearization method to solve Lane-Emden type equations. The solutions obtained using Matlab software are presented graphically and in tables. In Chapter 3, a review of the literature on the Bratu problem is given. We then use the spectral quasilinearization method and multi-domain spectral quasilinearization method to solve the Bratu problem. A comparison of the performance of the methods is shown. We use the bvp4c Matlab routine to validate the accuracy of the approximate solution. We further compare our solution with the existing literature and the ODE 45 routine, which uses the fourth-order Runge-Kutta method. In Chapter 4, we present the solution to the Troesch problem using the finite difference method and the higher-order spectral quasilinearization method. In Chapter 5, we present conclusions.

Chapter 2

The Lane-Emden Equation

In this chapter, we consider the solution of Lane-Emden type equations, ranging from simple to complex. The equations are solved separately using the finite difference method and the multi-domain spectral quasilinearization method. We present approximate solutions and compare the two methods based on the error given by each method.

2.1 The Lane-Emden equation

The Lane-Emden equation has the form

$$y''(x) + \frac{2}{x}y'(x) + g(x, y) = f(x), \tag{2.1}$$

where $g(x, y)$ is nonlinear. The equation is subject to the conditions

$$y(a) = \alpha, \quad y'(a) = \beta. \tag{2.2}$$

The Lane-Emden equation is a second-order nonlinear ordinary differential equation used to describe the structure of a polytropic gas sphere at equilibrium. Lane introduced the equation in 1870 with the objective of calculating the temperature and the density of the mass on the sun's surface [77]. The problem was later studied by Emden [78]. The Lane-

Emden equation describes the density near the surface of a gaseous star in astrophysics [79]. The Lane-Emden equation is an ordinary differential equation with a polytropic index n [79]. In astrophysics, the Lane-Emden equation is vital when the polytropic index n lies between 0 and 5. Exact solutions for the Lane-Emden equation can be obtained only for the polytropic index $n = 0, 1$, or 5. This is a major disadvantage for the practical use of the Lane-Emden equation in studies of the stellar structure, stability and oscillations [80]. Consequently many researchers have solved the Lane- Emden equation numerically. Adibi and Rismani [81] obtained the approximate solution using the Legendre-spectral method with Legendre-Gauss points as collocation points. Mirza [82] investigated the isothermal Lane-Emden equation by using the fractional approximation technique. Murkherjee et al. [79] investigated the Lane-Emden equation for different cases of polytropic index $n = 0, 1, 2, 3, 4, 5$, and confirmed that a closed-form is obtained only for $n = 0, 1$, and 5, while an infinite series was obtained for the cases of $n = 2, 3, 4$.

Taghavi and Pearce [83] solved the problem using the tau, Garlekin, and collocation methods. Van Gorder [84] solved the Lane-Emden equation using the δ-expansion method. Motsa and Shateyi [85] proposed a successive linearization method to solve the Lane-Emden problem. They found that the successive linearization method was easy to implement, and gave reliable and accurate solutions. Bhrawry and Alofi [86] proposed a shifted Jacobi-Gauss collocation method to solve the nonlinear Lane-Emden equation. They found the method to be simple and accurate and noted that even for few collocation points, the numerical results are excellent. Boubaker and Van Gorder [87] applied the Boubaker polynomials expansion scheme to two types of Lane-Emden problems; the first being for a polytropic gas sphere and the second for an isothermal gas sphere. Dehghan and Shakeri [88] investigated the solutions of the Lane-Emden equation using the variational iteration method (VIM). The VIM was applied successfully to the initial value problem (IVP) giving accurate numerical solutions. Doha et al. [89] presented a new second kind Chebyshev algorithm to solve the Lane-Emden equation, where the differential equation and its initial conditions were reduced to a system of algebraic equations. They found the algorithm to be simple and showed that it gave accurate solutions. Kumar et al. [90] demonstrated the efficiency of a numerical method they had applied to both the linear and nonlinear Lane-Emden equation using the Bernstein polynomial operational matrix of integration. Iqbal and Javed [91] applied and checked the performance of the optimal asymptotic method for solving Lane-Emden equation. Smarda and Khan [92] presented a new approach with an improved differential transformation method, which was applied successfully to obtain the solutions for different classes of the Lane-Emden equations. Rismani and Monfared [93] proposed an improved Legendre-spectral method to solve the initial value Lane-Emden equation. They used Legendre-Gauss points for collocation nodes and Legendre interpolation. They aimed to overcome the complexity of the singularity of the Lane-Emden equation at $x = 0$. In their work, they found solutions when $n = 2, 3, 4$ where there are no exact solutions.

We give below a brief discussion of the multi-domain spectral quasilinearization method in relation to its application to the Lane-Emden equation. Consider the initial value differential equation

$$y'' + \frac{2}{x}y' + g(x, y) = 0, \quad y(0) = \alpha_0, \quad y'(0) = \beta_0,$$

where the nonlinear function is represented by the function $g(x, y)$. The ordinary differential equation is reduced to a first order IVP system by setting $y' = f$. This gives,

$$\begin{aligned} &y' = f, \quad y(0) = \alpha_0, \\ &f' + \frac{2}{x}f + g(x, y) = 0, \quad f(0) = \beta_0. \end{aligned}$$

The multi-domain spectral quasilinearization method is applied in the first interval of the solution, and we get

$$\begin{aligned} &\frac{dy^{(1)}}{dx} = f^{(1)}, \quad y^{(1)}(0) = \alpha_0, \\ &\frac{df^{(1)}}{dx} + \frac{2}{x}f^{(1)} + g(x, y^{(1)}) = 0, \quad f^{(1)}(0) = \beta_0. \end{aligned}$$

An iterative scheme is developed and is used to decouple the system of equations to give

$$\begin{aligned}
&\frac{dy_{r+1}^{(1)}}{dx} = f_r^{(1)}, \quad y_{r+1}^{(1)}(0) = \alpha_0,\\
&\frac{df_{r+1}^{(1)}}{dx} + \frac{2}{x} f_{r+1}^{(1)} = -g(x, y_{r+1}^{(1)}), \quad f_{r+1}^{(1)}(0) = \beta_0.
\end{aligned}$$

In general, for $i = 1, 2, 3, \cdots$, the system of equations to be solved is given by

$$\begin{aligned}
&\frac{dy_{r+1}^{(i)}}{dx} = f_r^{(i)}, \quad y_{r+1}^{(i)}(x_{i-1}) = \alpha_{i-1},\\
&\frac{df_{r+1}^{(i)}}{dx} + \frac{2}{x} f_{r+1}^{(i)} = -g(x, y_{r+1}^{(i)}), \quad f_{r+1}^{(i)}(x_{i-1}) = \beta_{i-1}.
\end{aligned}$$

The collocation method is applied, giving

$$\begin{aligned}
&\sum_{k=0}^{N} \mathbf{D}_{j,k}^{(1)} y_{r+1}^{(1)}(x_k) = f_r^{(1)}(x_j), \quad y_{r+1}^{(1)}(x_N) = \alpha_0,\\
&\sum_{k=0}^{N} \mathbf{D}_{j,k}^{(1)} f_{r+1}^{(1)}(x_k) + \frac{2}{x_j} f_{r+1}^{(1)}(x_j) = -g(x_j, y_{r+1}^{(1)}(x_j)), \quad f_{r+1}^{(1)}(x_N) = \beta_0.
\end{aligned}$$

The above sums at $k = N$ including the boundary condition $y_{r+1}^{(1)}(x_N) = \alpha_0$ and $f_{r+1}^{(1)}(x_N) = \beta_0$, gives

$$\begin{aligned}
&\sum_{k=0}^{N-1} \mathbf{D}_{j,k}^{(1)} y_{r+1}^{(1)}(x_k) = f_r^{(1)}(x_j) - \mathbf{D}_{j,N}^{(1)} \alpha_0,\\
&\sum_{k=0}^{N-1} \mathbf{D}_{j,k}^{(1)} f_{r+1}^{(1)}(x_k) + \frac{2}{x_j} f_{r+1}^{(1)}(x_j) = -g(x_j, y_{r+1}^{(1)}(x_j)) = -\mathbf{D}_{j,N}^{(1)} \beta_0.
\end{aligned}$$

In the matrix form, these equations are written as

$$\begin{aligned}
A_1^{(1)} \mathbf{Y}_{r+1}^{(1)} &= \mathbf{R}_1^{(1)},\\
A_2^{(1)} \mathbf{F}_{r+1}^{(1)} &= \mathbf{R}_2^{(1)},
\end{aligned}$$

where

$$A_1^{(1)} = \mathbf{D}^{(1)}, \;\; A_2^{(1)} = \mathbf{D}^{(1)} + diag\left(\frac{2}{\mathbf{x}}\right),$$

$$R_1^{(1)}(j) = -\mathbf{D}_{j,N}^{(1)} \alpha_0 + f_r^{(1)}(x_j), \;\; R_2^{(1)}(j) = -\mathbf{D}_{j,N}^{(1)} \beta_0 - g(x_j, y_{r+1}^{(1)}(x_j)).$$

The approximate values of $y^{(k)}(x)$ and $f^{(k)}(x)$ are obtained by iterations for $i = 0, 1, 2, \cdots$, beginning with an initial approximation.

A brief illustration of the finite difference scheme is as follows. Consider the second-order boundary value problem

$$y'' + p(x)y' + q(x)y = r(x), \qquad a < x < b, \tag{2.3}$$

$$y(a) = \alpha \qquad y(b) = \beta.$$

where $x \in [a, b]$, with $x_i = a + ih$ and $h = (b - a)/N$ being the grid points.

In this work the explicit finite difference scheme and the Crank-Nicholson scheme are used to obtain the numerical solutions.

In equation (2.3), we replace the derivatives of y with forward and central differences to get

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i)\frac{y_{i+1} - y_i}{h} + q(x_i)y_i = r(x_i), \qquad i = 1, 2, 3, \cdots .$$

Multiplying throughout by h^2 and rearranging terms yields

$$(1 + hp(x_i))y_{i+1} - (2 + hp(x_i) - h^2q(x_i))y_i + y_{i-1} = h^2r(x_i), \qquad i = 1, 2, 3, \cdots .$$

Transposing y_i and y_{i-1} to the right, we get

$$(x_i + 2h)y_{i+1} = (2 + hp(x_i) - h^2q(x_i))y_i - y_{i-1} + h^2r(x_i), \qquad i = 1, 2, 3, \cdots .$$

The solution is found by solving iteratively starting from the initial approximation.

2.2 Numerical Experiments

In this section, we solve examples of the Lane-Emden equation using the finite difference method and the multi-domain spectral quasilinearization method.

2.2.1 A simple nonlinear Lane-Emden equation

We begin by solving a simple nonlinear Lane-Emden equation using the finite difference method. The problem has previously been solved by other methods. Iacono and De Feline [94] used a perturbation method to provide an accurate solution for the problem. Kanth and Aruna [95] used the variational iteration method for the problem and found that the method gives good results. Consider the initial value problem [95]

$$y'' + \frac{2}{x}y' + y^5 = 0, \qquad y(0) = 1, \quad y'(0) = 0. \tag{2.4}$$

This equation has the exact solution

$$y(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}.$$

A numerical solution is found to equation (2.4) using the explicit finite difference method and the multi-domain spectral quasilinearization method. The solution will be compared to the exact solution. Firstly, we find a numerical solution for equation (2.4) using the finite difference method. The central difference scheme is used for the second derivative y'' and forward difference scheme is used for the first derivative y', to obtain

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \frac{2}{x_i}\frac{y_{i+1} - y_i}{h} + y_i^5 = 0, \qquad i = 1, 2, 3, \cdots$$

Hence,

$$y_{i+1} = \frac{(2x_i + 2h)y_i - x_i y_{i-1} - h^2 x_i y_i^5}{(x_i + 2h)}, \qquad y_0 = 1, \quad y_1 = y_0.$$

We also sue the multi-domain spectral quasilinearization method (MD-SQLM) to reduce equation (2.4) to a system of first order IVP by substituting $y' = f$, which gives

$$y' = f, \quad y(0) = 1, \tag{2.5}$$

so that the equation reduces to

$$f' = -\frac{2}{x} f - y^5, \quad f(0) = 0. \tag{2.6}$$

To obtain the solutions we follow the procedure outlined in section 2.1. The results are presented in Table 2.1 and 2.2 and in Figure 2.1 - 2.3 .

Table 2.1: Numerical results $y(x)$ for equation (2.4) by using the finite difference method and multi-domain spectral method, compared with the exact solution, with $N = 100$.

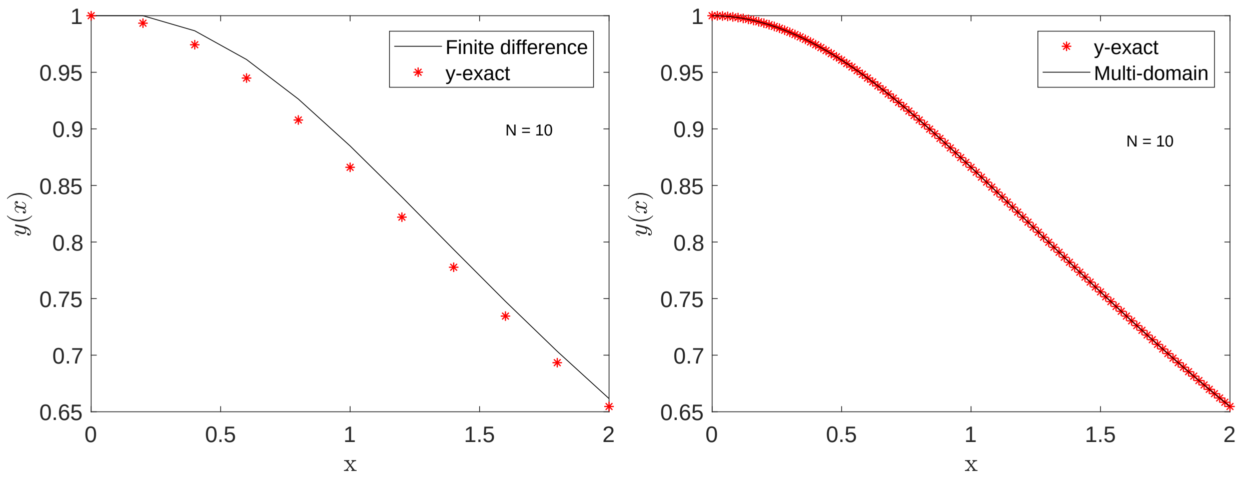
x	Exact	FDM	MD-SQLM
0.00	1.00000000000	1.00000000000	1.00000000000
0.20	0.99339926780	0.99404914976	0.99339926780
0.40	0.97435470369	0.97555759546	0.97435470369
0.60	0.94491118252	0.94650193716	0.94491118252
0.80	0.90784129900	0.90963053970	0.90784129900
1.00	0.86602540378	0.86783813095	0.86602540378
1.20	0.82199493653	0.82369294068	0.82199493653
1.40	0.77771377105	0.77920189588	0.77771377105
1.60	0.73455316031	0.73577509802	0.73455316031
1.80	0.69337524528	0.69430489260	0.69337524528

Table 2.1 shows the exact solutions and the approximate solution obtained using the finite difference and multi-domain methods. A comparison between the finite difference approximate solution and the exact solution shows they differ significantly whereas in the case of the multi-domain method shows results that are in agreement with the exact solution. This indicates of the accuracy of the multi-domain quasilinearization method in finding the solutions to the Lane-Emden equation. The finite difference method gives less accuracy in this instance.

Table 2.2: Absolute errors in the solution of equation (2.4) at different values of x with $N = 100$.

x	Error (FDM)	Error (MD-SQLM)
0.00	0.0000e+00	0.0000e+00
0.20	6.4988e-04	1.7097e-14
0.40	1.2029e-03	4.0190e-14
0.60	1.5908e-03	5.7176e-14
0.80	1.7892e-03	7.1942e-14
1.00	1.8127e-03	9.0705e-14
1.20	1.6980e-03	9.0705e-14
1.40	1.4881e-03	9.1815e-14
1.60	1.2219e-03	8.7041e-14
1.80	9.2965e-04	8.6819e-14

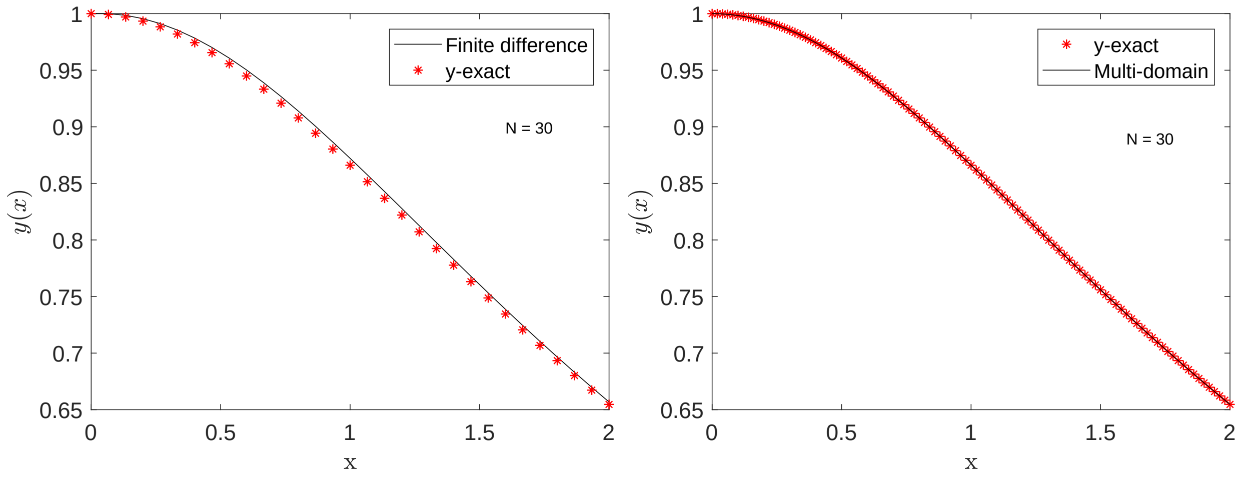
Table 2.2 shows the absolute error in the approximate solutions. The multi-domain spectral quasilinearization method has the smallest error of $1.7097e - 14$ which indicates the accuracy of the method, while the error in the finite difference is much greater, of the order $6.4988e - 04$. The comparison shows that the multi-domain spectral quasilinearization method is the better of the two methods for this form of the Lane-Emden equation.



(a) (b)

Figure 2.1: Comparison of exact and numerical results for equation (2.4) when N =10 for (a) the finite difference method, and (b) the multi-domain spectral method.

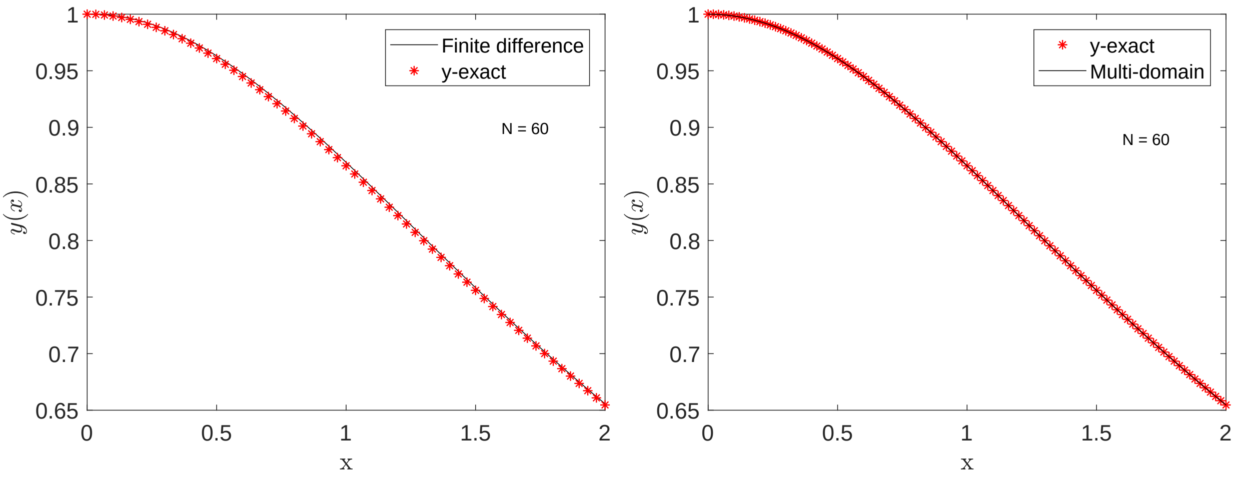
Figure 2.1 allows comparison between the approximate solution and analytical solution when $N = 10$. The solution from the finite difference method converges more slowly than that for the multi-domain spectral quasilinearization method, which converges rapidly to the exact solution.



(a) (b)

Figure 2.2: Comparison of exact and numerical results for equation (2.4) when N =30 for (a) the finite difference method, and (b) the multi-domain spectral method.

Comparing results in Figure 2.2 for $N = 30$, with those from $N = 10$, there is a slight improvement in convergence of the finite difference method, which indicates consistency. For the multi-domain spectral quasilinearization method, the solution has already converged and therefore the solution does not change. The method is stable.



(a)

(b)

Figure 2.3: Comparison of exact and numerical results when N =60 for (a) the finite difference method, and (b) the multi-domain spectral methods.

Figure 2.3 shows the solution as the number of grid points is increased to $N = 60$. The finite difference solution converges to the exact solution. The multi-domain spectral remains stable and unchanged.

2.2.2 A more complex nonlinear Lane-Emden equation

We now consider a slightly more complex type of the Lane-Emden equation, which we will solve using the finite difference and multi-domain spectral quasilinearization method. This problem is more complex because we dealing with exponential terms. It has already been solved by other methods. For example, Shiralashetti and Kumbinarasaiah [96] solved this problem using the Hermite wavelets operational matrix method. Bhrawry and Alofi [86] solved it using the Jacobi-Gauss collocation method, noting that their method was effective. Consider the initial value problem [86]

$$y'' + \frac{2}{x}y' + 4\left(2e^{y} + e^{\frac{y}{2}}\right) = 0, \qquad y(0) = 0, \quad y'(0) = 0. \tag{2.7}$$

The equation has the exact solution

$$y(x) = -2\ln\left(1 + x^2\right).$$

In our numerical solutions, we first introduce a finite difference scheme using the central difference for y'' and forward difference for y'. The equation becomes

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \frac{2}{x_i}\frac{y_{i+1} - y_i}{h} + 4\left(2e^{y_i} + e^{\frac{y_i}{2}}\right) = 0, \qquad i = 1, 2, 3, \cdots$$

hence,

$$(x_i + 2h)y_{i+1} = (2x_i + 2h)y_i - x_i y_{i-1} - 4h^2 x_i\left(2e^{y_i} + e^{\frac{y_i}{2}}\right), \qquad y_0 = 0, \quad y_1 = y_0.$$

When the multi-domain spectral quasilinearization method is applied to equation (2.7), we get a system of first order IVPs by substituting $y' = f$, which gives

$$y' = f, \quad y(0) = 0, \tag{2.8}$$

$$f' = -\frac{2}{x}f - 4\left(2e^{y} - e^{\frac{y}{2}}\right), \quad f(0) = 0. \tag{2.9}$$

Then we follow the procedure described in section 1.4.3. to obtain the solutions. The results are presented in Tables 2.3 and 2.4 and in Figure 2.4 - 2.5.

Table 2.3: Numerical results of equation (2.7) with $N = 100$ for the exact solution, the finite difference method and multi-domain spectral quasilinearization method.

x	Exact	FDM	MD-SQLM
0.00	0.00000000000	0.00000000000	0.00000000000
0.20	-0.07844142631	-0.07083538808	-0.07844142631
0.40	-0.29684001024	-0.28371786342	-0.29684001024
0.60	-0.61496939950	-0.59926445744	-0.61496939950
0.80	-0.98939248367	-0.97363195241	-0.98939248367
1.00	-1.38629436112	-1.37212891666	-1.38629436112
1.20	-1.78399607861	-1.77228956096	-1.78399607861
1.40	-2.17037853667	-2.16146157902	-2.17037853667
1.60	-2.53952108973	-2.53341323961	-2.53952108973
1.80	-2.88912653849	-2.88568499675	-2.88912653849

Table 2.3 shows the solution of equation 2.7 using the exact solution, the finite difference method, and the multi-domain spectral quasilinearization method and the exact solution. In the finite difference method at $x = 0.40$, the solution has converged to only two decimal places while for the multi-domain spectral quasilinearization method the solution has converged to the exact solution.

Table 2.4: Absolute errors in the solution of Equation (2.7) for different values of x when $N = 100$

x	Error (FDM)	Error (MD-SQLM)
0.00	0.0000e+00	0.0000e+00
0.20	7.6060e-03	4.9960e-16
0.40	1.3122e-02	1.1657e-15
0.60	1.5705e-02	7.4385e-15
0.80	1.5761e-02	1.1324e-14
1.00	1.4165e-02	1.5099e-14
1.20	1.1707e-02	3.3751e-14
1.40	8.9170e-03	6.2617e-14
1.60	6.1079e-03	8.0380e-14
1.80	3.4415e-03	9.5035e-14

Table 2.4 shows the absolute error for the finite difference method and the multi-domain spectral quasilinearization method. For $x = 1.60$, the finite difference method has an error of $6.1079e - 03$, while for the multi-domain spectral quasilinearization method the error is much smaller, at $8.0380e - 14$. The multi-domain spectral quasilinearization method is, thus, the more accurate of the two methods.

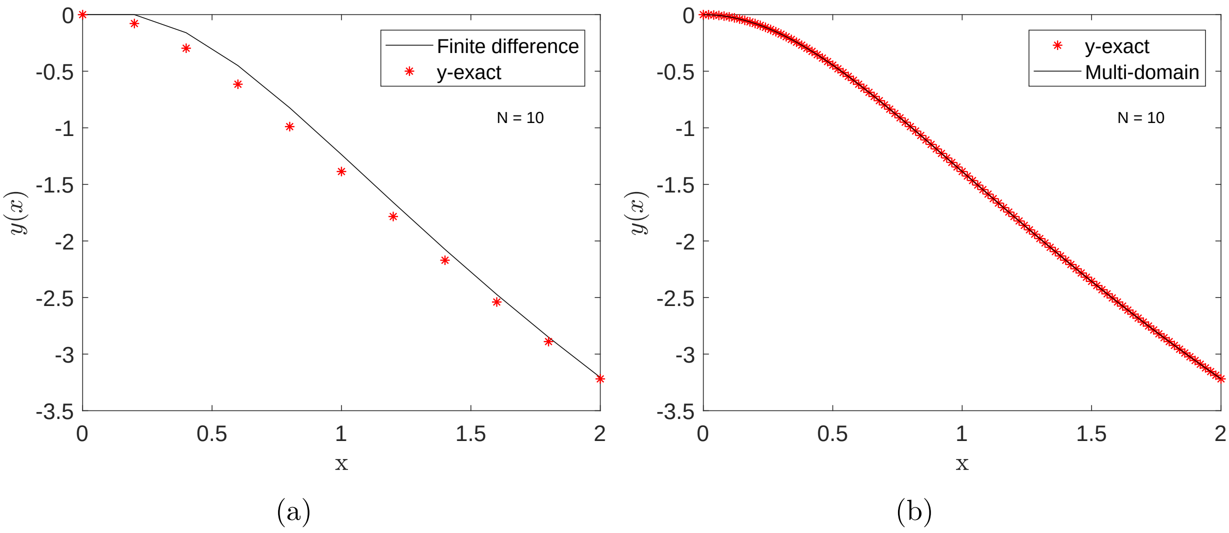
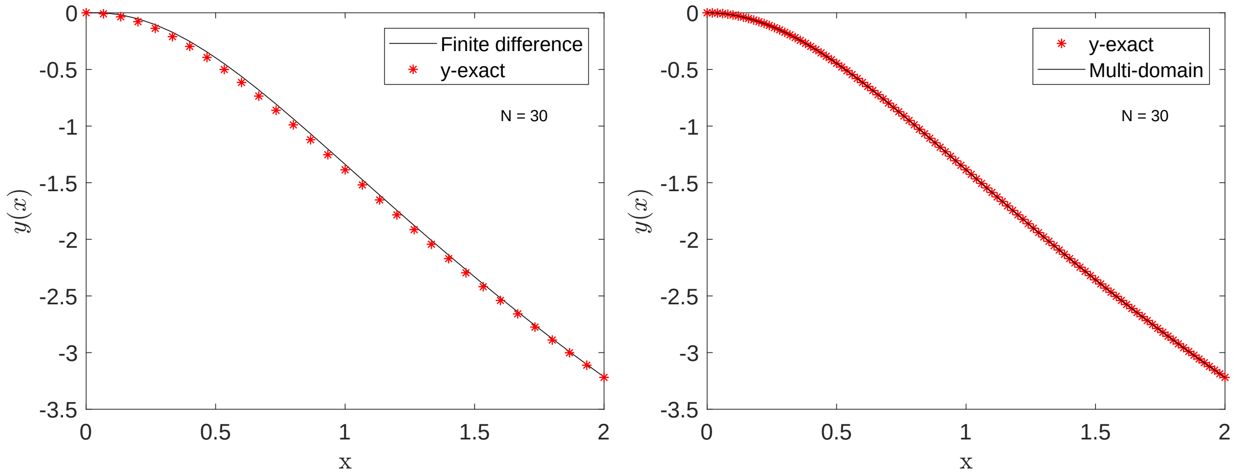


Figure 2.4: Comparison of exact and numerical results for equation (2.7) when N =10 for (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

Figure 2.4 shows the solutions when $N = 10$ found by the finite difference method and the multi-domain spectral quasilinearization method and compares them with the exact solution. The solution by the finite difference converges slowly whereas for the multi-domain spectral quasilinearization method, it converges faster when $N = 10$.



(a)

(b)

Figure 2.5: Comparison of exact and numerical results for equation (2.7) when N =30 from (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

Figure 2.5 shows the solutions found by the two numerical methods when $N = 30$. There is a slight improvement in the solution by the finite difference method from $N = 10$. There is no change in the multi-domain spectral quasilinearization method, which means the method is consistent and stable.

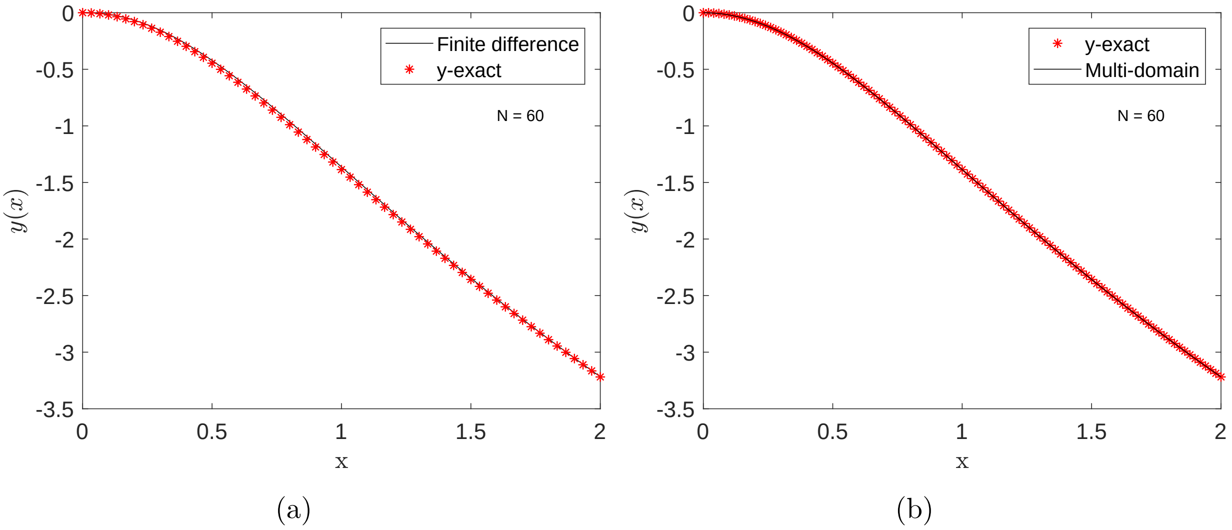


Figure 2.6: Comparison of exact and numerical results for equation (2.7) when N =60 for (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

Figure 2.6 shows the solution when the number of grid points is increased to $N = 60$. As with lower number of grid points, the finite difference method now converges to the exact solution and is consistent. The multi-domain spectral quasilinearization method converges and is stable. The multi-domain spectral quasilinearization method may, however, be a more preferable method to use to find the solution.

2.2.3 An even more complex Lane-Emden equation

We now extend the complexity of the Lane-Emden equation that we will solve. This problem which include logarithmic function terms, was solved by Bhrawry and Alofi [86] using the Jacobi-Gauss collocation method. Consider an initial value problem [86]

$$y'' + \frac{2}{x}y' - 6y - 4y\ln y = 0, \qquad y(0) = 1, \quad y'(0) = 0, \tag{2.10}$$

with a closed form solution

$$y(x) = e^{x^2}.$$

When the solution is obtained by finite difference schemes, equation (2.10) becomes

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \frac{2}{x_i}\frac{y_{i+1} - y_i}{h} - 6y_i - 4y_i \ln y_i = 0, \qquad i = 1, 2, 3, \ldots$$

hence,

$$(x_i + 2h)y_{i+1} = (2x_i + 2h)y_i - x_i y_{i-1} - h^2 x_i (6y_i + 4y_i \ln y_i), \qquad y_0 = 0, \quad y_1 = y_0.$$

We then use the multi-domain spectral quasilinearization method to find the solution. We begin by reducing equation (2.10) to a system of first order initial value problem by substituting $y' = f$, which gives

$$y' = f, \quad y(0) = 1, \tag{2.11}$$

$$f' = -\frac{2}{x} f + 6y + 4y \ln y, \quad f(0) = 0. \tag{2.12}$$

We follow the procedure described in Section 1.4.3. to obtain the solution.

Results are presented in Table 2.5 and Figure 2.7, 2.8, and 2.9.

Table 2.5: Numerical results $y(x)$ of equation (2.10) using the exact solution, the finite difference method and multi-domain spectral quasilinearization method with $N = 100$.

x	Exact	FDM	MD-SQLM
0.00	1.00000000000	1.00000000000	1.00000000000
0.20	1.04081077419	1.03660154636	1.04081077419
0.40	1.17351087099	1.16368196422	1.17351087099
0.60	1.43332941456	1.41435368519	1.43332941456
0.80	1.89648087930	1.86085484710	1.89648087930
1.00	2.71828182846	2.65012977529	2.71828182846
1.20	4.22069581700	4.08528016832	4.22069581700
1.40	7.09932706516	6.81700947288	7.09932706516
1.60	12.93581731554	12.31422128415	12.93581731554
1.80	25.53372174735	24.08153759156	25.53372174735

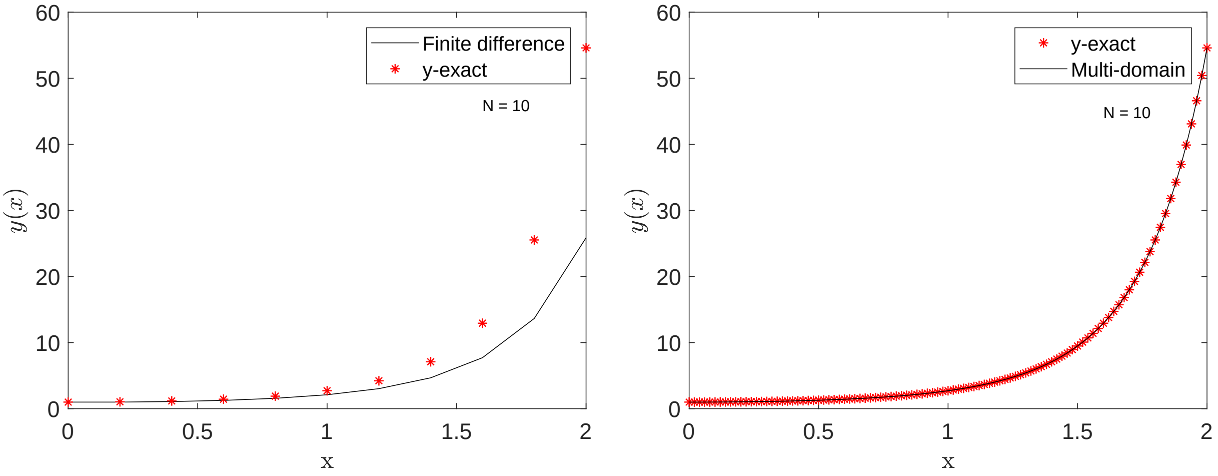
Table 2.5 shows the exact solutions and the approximate solutions obtained using the finite difference and the multi-domain spectral quasilinearization methods. A comparison of these results indicates that the finite difference solution differs slightly from the exact solution but for the multi-domain spectral quasilinearization method the solution converges

to the exact solution. The comparison, therefore, shows that the multi-domain method is the better numerical method to solve this example of the Lane Emden equation.

Table 2.6: Absolute errors in the solution of Equation (2.10) at different values of x with $N = 100$

x	Error (FDM)	Error (MD-SQLM)
0.00	0.0000e+00	0.0000e+00
0.20	4.2092e-03	5.5511e-15
0.40	9.8289e-03	1.4655e-14
0.60	1.8976e-02	3.7748e-15
0.80	3.5626e-02	1.5987e-14
1.00	6.8152e-02	2.4425e-14
1.20	1.3542e-01	8.4377e-14
1.40	2.8232e-01	2.1316e-13
1.60	6.2160e-01	4.5652e-13
1.80	1.4522e+00	1.2470e-12

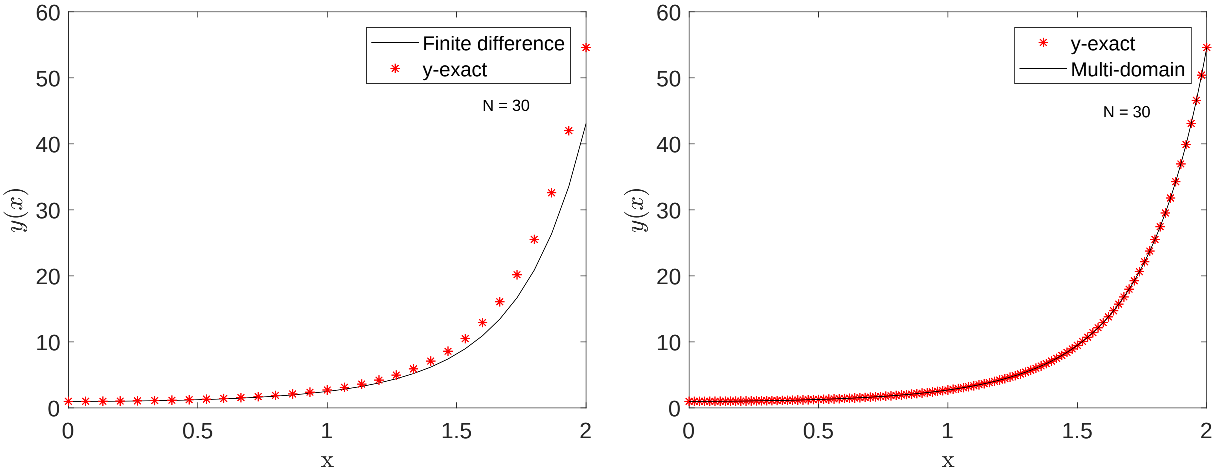
Table 2.6 compares the absolute errors obtained using the finite difference and multi-domain spectral quasilinearization methods. It can be seen that the multi-domain spectral quasilinearization method has the smallest error of $5.5511e-15$, which indicates the accuracy of the method. The error in the finite difference is significantly larger and of the order $4.2092e-03$. The comparison shows that the multi-domain spectral quasilinearization method is, in this case, the better method to use to solve the Lane-Emden equation.



(a) (b)

Figure 2.7: Comparison of exact and numerical results for equation (2.10) when N =10 for (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

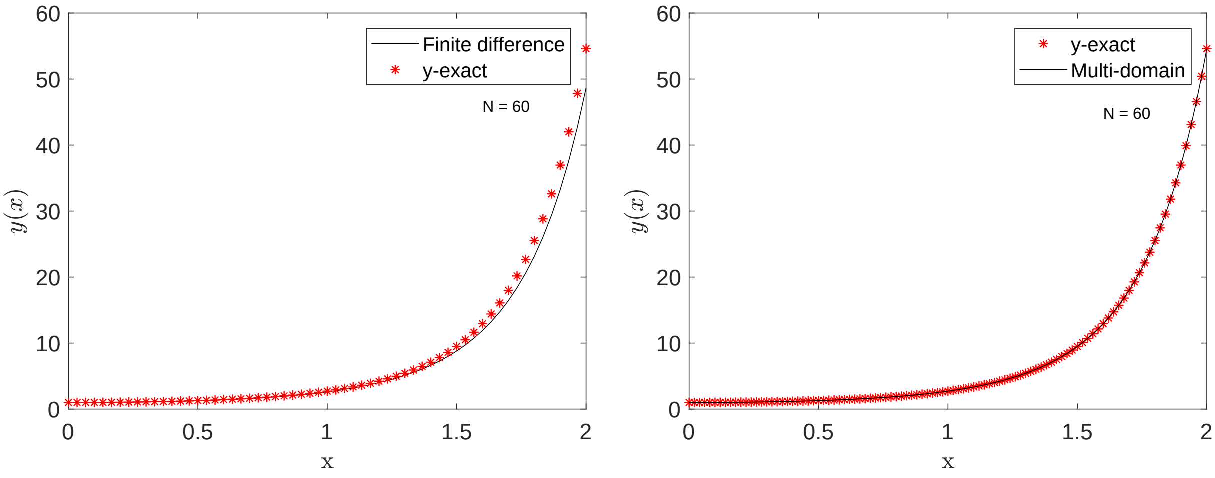
Figure 2.7 shows the results obtained using the finite difference method and multi-domain spectral quasilinearization method, compared with those for the exact solution. To start, we picked a small value of grid points, $N = 10$ to observe the performance of the methods. The finite difference method converges more slowly than does the multi-domain spectral quasilinearization method, which converges to the exact solution.



(a) (b)

Figure 2.8: Comparison of exact and numerical results for equation (2.10) when N =30 for (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

Figure 2.8 shows the comparison of the two methods when the number of grid points is increased so $N = 30$. For the finite difference method, the larger number of grid points means there is improved convergence to the exact solution. The finite difference is consistent. The multi-domain method retains its accuracy even when the grid points are increased, and it is stable.



(a)

(b)

Figure 2.9: Comparison of exact and numerical results for equation (2.10) when N =60 for (a) the finite difference, and (b) the multi-domain spectral quasilinearization methods.

Figure 2.9 shows the solutions as the grid points are increased with $N = 60$. In the finite difference method the solution converges to the exact solution. In the multi-domain spectral quasilinearization method still shows accurate results even when the grid points are increased to $N = 60$. Bacause of the fast convergence, the multi-domain spectral quasilinearization method is a much better method to use for this equation.

2.3 Summary

In this chapter we presented the solution of three Lane-Emden type equations, ranging from simple to highly nonlinear. The finite difference method and multi-domain spectral quasilinearization method were used to obtain the approximate solutions. We have showed that, for all three equations, the multi-domain spectral quasilinearization method gives faster convergence and better accuracy than does the finite difference . In the next chapter, we present the solution for the Bratu problem using the spectral quasilinearization method, and we will compare it with the performance of the higher-order spectral quasilinearization method.

Chapter 3

The Bratu Problem

In this chapter, we present the numerical solution of the Bratu problem which is a nonlinear boundary value problem. The solution methods are the spectral quasilinearization method and a higher-order spectral quasilinearization method. The results obtained are compared with the known exact solution and the convergence of the methods is determined.

3.1 The Bratu equation

Consider the nonlinear Bratu problem

$$y''(x) + f(x, y) = 0, \tag{3.1}$$

where $f(x, y)$ is nonlinear. The equation is subject to boundary conditions

$$y(a) = \alpha, \quad y(b) = \beta. \tag{3.2}$$

The Bratu equation has been studied in engineering and physics as a fuel ignition model for chemical and thermal reactions, thermal combustion theory and radiative heat transfer [97]. The equation was first introduced by Bratu in 1914 [98]. The Bratu problem is an eigenvalue problem which is nonlinear and it is usually treated as a benchmark problem to test their accuracy of numerical methods [99]. The Bratu problem has a parameter

λ which describes the number of possible solutions; the problem has two solutions for $\lambda < \lambda_c$, one solution for $\lambda = \lambda_c$ and no solution for $\lambda > \lambda_c$, where λ_c is a critical turning point [100]. Ragb et al. [101] introduced a numerical scheme based on the differential quadrature method for solving the Bratu problem. In this method, the unknowns were approximated using the differential quadrature approximation, which led to a nonlinear algebraic system that was solved iteratively. These schemes were successful in solving the nonlinear Bratu problem with few grid points and the results agreed with the 1-D and 2-D closed forms. The moving least square differential quadrature method provided rapid convergence compared to the differential quadrature method. Caglar et al. [102] used the B-Spline method to obtain the approximate solution for the Bratu problem, for different values of λ at $\lambda = 1$, 2, and 3.51. A comparison was made between their numerical results and those from other methods in the literature. They found their results were accurate at $\lambda = 1$, but as λ increased the error increased. They concluded that the B-Spline method was effective and more accurate than the Laplace method and the decomposition method. Hassan and Semary [103] presented analytic approximate solutions for different λ for the Bratu problem using the homotopy analysis method. They used the averaged residual error to find the optimal value of the convergence-controller parameter h. In their method, the nonlinear terms were replaced by the Taylor series, and they achieved high accuracy in their results. Temimi and Ben-Romdhane [104] proposed a new iterative scheme based on the Newton-Raphson-Kantorovich approximate method to solve the Bratu problem. A comparison was performed between B-Spline and Laplace transform methods already in the literature. The scheme produced accurate and efficient results and was far more reliable and accurate than the B-Spline and Laplace transform methods. Odejide and Aregbesola [105] examined the nonlinear Bratu problem in two dimensions. The equation was solved using three methods, the finite difference method, the weighted residual method, and an analytical method. They obtained accurate results for all three methods. Ghomanjani and Shateyi [106] proposed a new approach for the Bratu problem, which was based on the Bernstein polynomial approximations. When compared with methods in the existing literature their method was found to be accurate and efficient. Tomar and Pandey [107] introduced a numerical technique to solve the Bratu problem, which employed the quasilinearization method to linearize the problem. The linearized

Bratu problem was then solved using an optimal Picard iterative method. They found that their method was efficient and accurate. Venkatesh et al. [108] solved the initial value Bratu problem, using the Legendre wavelet method to obtain the solution. They compared the approximate solution and the exact solution and showed that their method was highly accurate and reliable. Saravi et al. [109] solved the Bratu problem using the variational iteration method. They found that the method was effective and reliable.

3.2 Numerical Experiments

In this section, we solve the Bratu equation using the spectral quasilinearization method (SQLM) and higher-order spectral quasilinearization method (HO-SQLM).

3.2.1 Example 1

Consider the simple problem [104]

$$y''(x) + \lambda e^{y(x)} = 0, \qquad y(0) = 0, \quad y(1) = 0, \quad 0 \leq x \leq 1, \tag{3.3}$$

we begin by using the spectral quasilinearization method. Linearizing the equations using the SQLM scheme gives

$$a_{2,r}y''_{r+1} + a_{1,r}y'_{r+1} + a_{0,r}y_{r+1} = R_r, \tag{3.4}$$

with boundary conditions

$$\begin{aligned} y_{r+1} &= 0, \quad \text{at} \quad x = 0, \\ y_{r+1} &= 0, \quad \text{at} \quad x = 1, \end{aligned} \tag{3.5}$$

where

$$a_{2,r} = 1, \quad a_{1,r} = 0, \quad a_{0,r} = \lambda e^{y_r}, \tag{3.6}$$

$$R_r = \lambda e^{y_r} y_r - \lambda e^{y_r}. \tag{3.7}$$

Before applying the spectral collocation method, the domain $x \in [0, 1]$ is transformed to $z \in [-1, 1]$ using the linear transformation $x = \dfrac{(z+1)}{2}$

To transform the system (3.37) so as to be defined in $[-1, 1]$, it is enough to transform the derivatives using the chain rule. The scaled differentiation matrix in the interval $[0, 1]$ is given by $\mathbf{D} = 2D$ where D is the standard differentiation matrix. Thus, at the collocation points $z_i = \cos\left(\frac{\pi i}{N}\right)$ for $i = 0, 1, 2, \ldots, N$, the derivatives u'' become

$$y''(x_i) \rightarrow \mathbf{D}^2\mathbf{y}, \quad i = 0, 1, 2, \ldots, N.$$

Applying spectral collocation on (3.37) gives

$$\mathbf{D}^2\mathbf{y}_{r+1} + a_{1,r}\mathbf{D}\mathbf{y}_{r+1} + a_{0,r}\mathbf{y}_{r+1} = \mathbf{R}_r, \tag{3.8}$$

which can be written as

$$\left[\mathbf{D}^3 + a_{1,r}\mathbf{D} + a_{0,r}\right]\mathbf{y}_{r+1} = \mathbf{R}_r. \tag{3.9}$$

In matrix form, equation (3.9) can be written as

$$A\mathbf{F} = \boldsymbol{R_r} \tag{3.10}$$

where

$$A = \mathbf{D}^2 + a_{1,r}\mathbf{D} + a_{0,r}\mathbf{I}. \tag{3.11}$$

The boundary conditions (3.38) are

$$y_{r+1}(z_N) = 0, \tag{3.12}$$

$$y_{r+1}(z_0) = 0. \tag{3.13}$$

The boundary conditions corresponding to y are imposed as equations on the first and last two rows of the matrix A and vector R_r.

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ A_{1,0} & A_{1,1} & A_{1,2} & \cdots & A_{1,N} \\ A_{2,0} & A_{2,1} & A_{2,2} & \cdots & A_{2,N} \\ A_{3,0} & A_{3,1} & A_{3,2} & \cdots & A_{3,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{N-2,0} & A_{N-2,1} & A_{N-2,2} & \cdots & A_{N-2,N} \\ A_{N-1,0} & A_{N-1,1} & A_{N-1,2} & \cdots & A_{N-1,N} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_{r+1,0} \\ y_{r+1,1} \\ y_{r+1,2} \\ y_{r+1,3} \\ \vdots \\ y_{r+1,N-2} \\ y_{r+1,N-1} \\ y_{r+1,N} \end{bmatrix} = \begin{bmatrix} 0 \\ R_{r,1} \\ R_{r,2} \\ R_{r,3} \\ \vdots \\ R_{r,N-2} \\ R_{r,N-1} \\ 0 \end{bmatrix}$$

The approximate solutions $y(x)$ are obtained by solving the matrix system A iteratively for $r = 0, 1, 2, \ldots$, beginning with a suitable initial guess. The chosen initial guess is a simple function such as a polynomial which satisfies the boundary conditions. We use the following initial guess

$$y_0(x) = 0.$$

We next show the solution for the higher-order spectral quasilinearization method (HO-SQLM). Consider equation (3.3), for which the nonlinear component is given as

$$N = \lambda e^y,$$

and approximated as

$$N \approx \lambda e^{y_i} + \alpha_{0,i}(y - y_i) + H[y, y_i],$$

where

$$\alpha_{0,i} = \lambda e^{y_i}.$$

Then we follow the procedure given in section (1.4.3.).

The solution of the problem is presented in Tables 3.1-3.5 and Figures 3.1, 3.2.

Table 3.1: Numerical results $y(x)$ of Equation (3.3) for the SQLM and HO-SQLM for $N = 30$

x	Exact	SQLM	ho-SQLM
0.00	0.00000000000	0.00000000000	0.00000000000
0.17	0.60168967660	0.60124335136	0.60167522395
0.25	0.84674416250	0.84609447188	0.84672312460
0.35	1.05244530281	1.05161294186	1.05241834979
0.45	1.17115790873	1.17021454905	1.17112736142
0.55	1.17115790873	1.17021454905	1.17112736142
0.65	1.05244530281	1.05161294186	1.05241834979
0.79	0.72605577486	0.72550804956	0.72603803874
0.87	0.47927323695	0.47892335005	0.47926190712
0.99	0.04349197304	0.04346183446	0.04349099711
1.00	0.00000000000	0.00000000000	0.00000000000

Table 3.1 shows the numerical solutions obtained using the spectral quasilinearization method and the higher-order spectral quasilinearization method. The results show that although neither set of solutions match the exact solution, those obtained by the HO-SQLM are closer than those obtained using SQLM. For example, when $x = 0.17$ the results agree for the SQLM to four decimal places, while for the HO-SQLM they agree to five decimal places.

Table 3.2: Absolute error in solving equation (3.3) using the SQLM and HO-SQLM when $N = 30$

x	SQLM	HO-SQLM
0.00	0.0000e+00	0.0000e+00
0.17	4.4633e-04	1.4453e-05
0.25	6.4969e-04	2.1038e-05
0.35	8.3236e-04	2.6953e-05
0.45	9.4336e-04	3.0547e-05
0.55	9.4336e-04	3.0547e-05
0.65	8.3236e-04	2.6953e-05
0.79	5.4773e-04	1.7736e-05
0.87	3.4989e-04	1.1330e-05
0.99	3.0139e-05	9.7593e-07
1.00	0.0000e+00	0.0000e+00

Table 3.2 shows the absolute error for the SQLM and HO-SQLM. For $x = 0.87$ the error for the SQLM is $3.4989e - 04$ while the HO-SQLM performs only marginally better, with an error of $1.1330e - 05$. Together with results from Table 3.3, this indicates that the HO-SQLM is the slightly more accurate of the two methods for solving the Bratu problem.

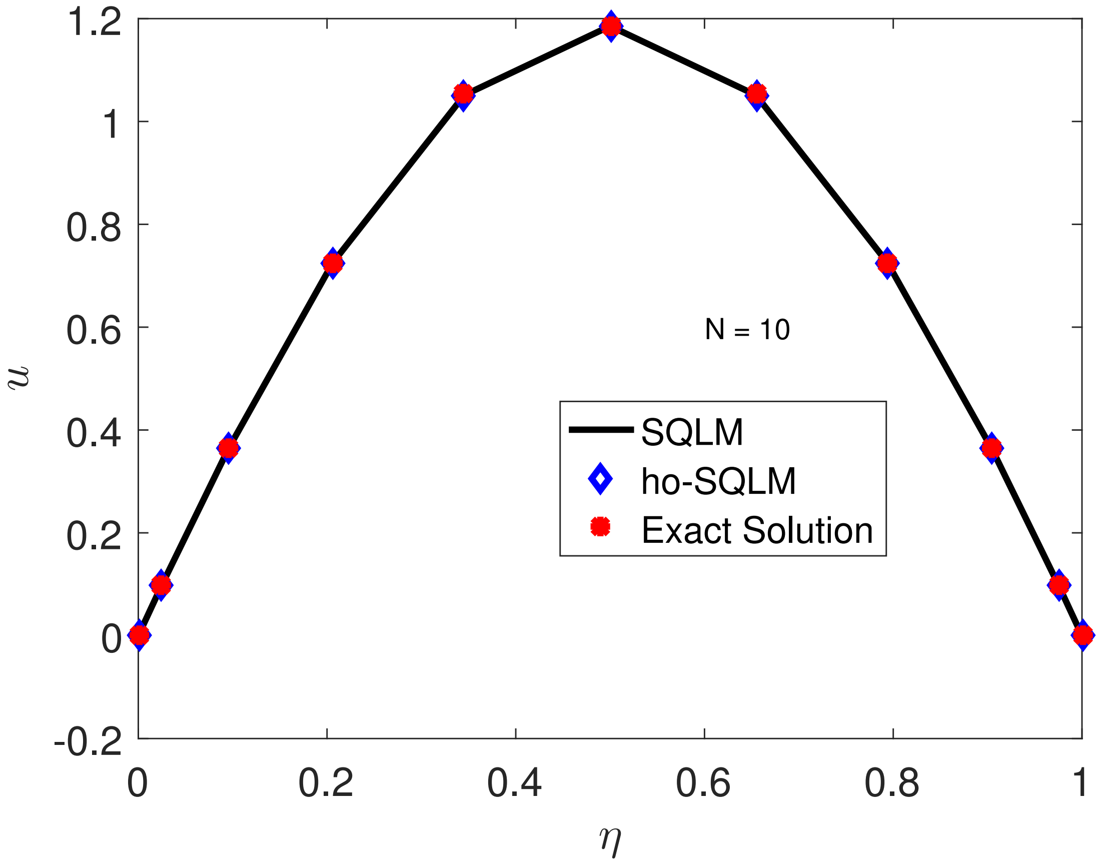


Figure 3.1: Numerical results of the SQLM and the HO-SQLM for equation (3.3) when $N = 10$.

Figure 3.1 shows a graphical comparison of the solution for the SQLM, HO-SQLM and the exact solution. Both methods appear to be accurate and give reliable results.

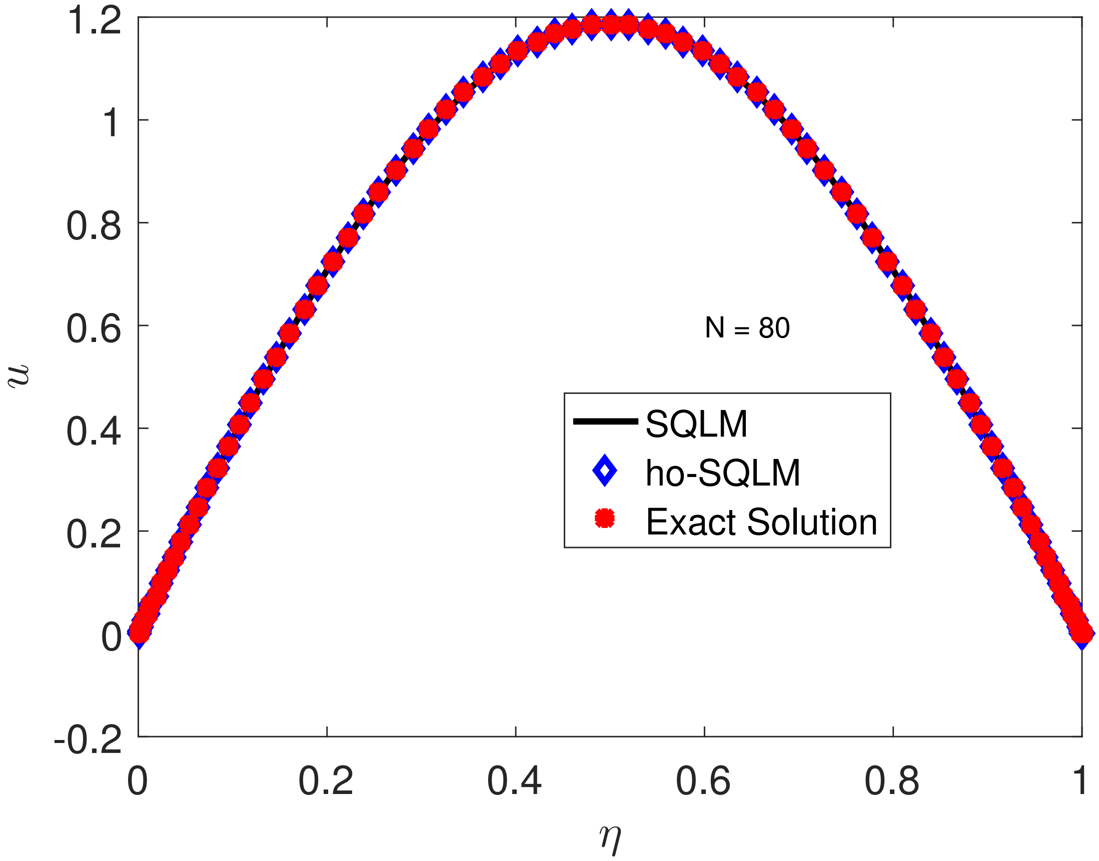


Figure 3.2: Numerical results of the Bratu equation using the SQLM and the HO-SQLM of equation (3.3) when $N = 80$.

Figure 3.2 shows a comparison of the solution of the Bratu equation using the SQLM, HO-SQLM and the exact solution, when N has been increased from 10 to 80. In this case the solution curves are a good match. The results indicate that, when the number of grid points is increased to 80, further improvements in accuracy is achieved for both methods.

Table 3.3: Numerical results of equation (3.3) using HO-SQLM in comparison with MATLAB inbuilt routine bvp4c for $\lambda = 0.1$.

x	$y(x)$	bvp4c
0.0	0.00000000	0.00000000
0.1	0.00454150	0.00454022
0.2	0.00807854	0.00807610
0.3	0.01060755	0.01060420
0.4	0.01212597	0.01212204
0.5	0.01263229	0.01262815
0.6	0.01212597	0.01212204
0.7	0.01060755	0.01060420
0.8	0.00807854	0.00807610
0.9	0.00454150	0.00454022
1.0	0.00000000	0.00000000

Table 3.3 shows the results of equation (3.3) where $\lambda = 0.1$. The results were obtained using the HO-SQLM and MATLAB bvp4c inbuild routine. The results are in agreement up to five decimal places.

Table 3.4: Numerical results of equation (3.3) in comparison with MATLAB bvp4c solver for $\lambda = 1$

x	HO-SQLM	bvp4c
0.0	0.00000000	0.00000000
0.1	0.04984679	0.04984672
0.2	0.08918993	0.08918980
0.3	0.11760910	0.11760891
0.4	0.13479025	0.13479003
0.5	0.14053921	0.14053898
0.6	0.13479025	0.13479003
0.7	0.11760910	0.11760891
0.8	0.08918993	0.08918980
0.9	0.04984679	0.04984672
1.0	0.00000000	0.00000000

Table 3.4 show the results for $\lambda = 1$ obtained by the HO-SQLM and MATLAB bvp4c inbuild routine. The results are in agreement for up to seven decimal places.

Table 3.5: Comparison of the residual error in the HO-SQLM and SQLM for $\lambda = 0.1$.

iter.	HO-SQLM	SQLM
1	1.13178911e-12	2.93089749e-03
2	1.13178911e-12	3.84472562e-09
3	1.13178911e-12	1.06059606e-12
4	1.13178911e-12	1.64884773e-12
5	1.13178911e-12	1.13178911e-12
6	1.13178911e-12	1.13178911e-12
7	1.13178911e-12	1.13178911e-12
8	1.13178911e-12	1.13178911e-12
9	1.13178911e-12	1.13178911e-12
10	1.13178911e-12	1.13178911e-12

Table 3.5 shows the residual error of the SQLM and HO-SQLM when $\lambda = 0.1$. The higher-order spectral quasilinearization method is seen to be more accurate than the spectral quasilinearization method.

3.2.2 Example 2

In this section we consider a more complex problem. The problem is again solved using both the spectral quasilinearization method and the higher-order spectral quasilinearization method. This problem is a highly nonlinear initial value problem.

In this example, we solve the equation (see [110])

$$y''(t) + y(t) + ay^3(t) + b\sin(y(t)) = 0, \qquad y(0) = \frac{\pi}{18}, \quad y'(0) = 0. \tag{3.14}$$

We begin by linearizing the equations using the SQLM scheme, giving

$$a_{2,r}y''_{r+1} + a_{1,r}y'_{r+1} + a_{0,r}y_{r+1} = R_r, \tag{3.15}$$

with boundary conditions

$$\begin{aligned} y_{r+1} &= 0, \ \ \text{at} \ \ x = \frac{\pi}{18}, \\ y'_{r+1} &= 0, \ \ \text{at} \ \ x = 0, \end{aligned} \tag{3.16}$$

where

$$a_{2,r} = 1, \ \ a_{1,r} = 0, \quad a_{0,r} = 1 + 3ay_r^2 + b\cos(y_r), \tag{3.17}$$

$$R_r = 2ay_r^3 + by\cos(y_r) - b\sin(y). \tag{3.18}$$

The resulting solution is of the form

$$A\mathbf{F} = \boldsymbol{R_r} \tag{3.19}$$

where

$$A = \mathbf{D}^2 + a_{1,r}\mathbf{D} + a_{0,r}\mathbf{I}. \tag{3.20}$$

The boundary conditions are

$$y_{r+1}(z_N) = \frac{\pi}{18}, \tag{3.21}$$

$$y'_{r+1}(z_N) = 0. \tag{3.22}$$

Now, for comparison, we use the HO-SQLM to obtain the solution. The nonlinear component is given as

$$N = ay^3 + b\sin(y), \tag{3.23}$$

which can be approximated as

$$N[y_i] \approx ay_i^3 + b\sin(y_i) + \alpha_{0,i}(y - y_i) + H[y, y_i], \tag{3.24}$$

where

$$H[y, y_i] = ay^3 + b\sin(y) - \left(ay_i^3 + b\sin(y_i)\right) - \alpha_{0,i}\left(y - y_i\right), \tag{3.25}$$

and

$$\alpha_{0,i} = 3ay_i^2 + b\cos(y_i). \tag{3.26}$$

Equation (3.14) then becomes

$$y'' + (1 + \alpha_{0,i})\, y + H[y, y_i] = \alpha_{0,i}y_i - \left(ay_i^3 + b\sin(y_i)\right). \tag{3.27}$$

Here $H[y, y_i]$ will be written as

$$H[y, y_i] = H[y_r, y_i] + \beta_{0,r}\left(y_{r+1} - y_r\right), \tag{3.28}$$

where

$$\beta_{0,r} = 3ay_r^2 + b\cos(y_r) - \alpha_{0,i}, \tag{3.29}$$

so that when we collocate, we have

$$y''_{r+1} + (1 + \alpha_{0,i})\, y_{r+1} + H[y_r, y_i] + \beta_{0,r}\, (y_{r+1} - y_r) = \phi, \tag{3.30}$$

$$\tag{3.31}$$

where

$$\phi = \alpha_{0,i} - \left(a y_i^3 + b \sin(y_i) \right). \tag{3.32}$$

We now write this equation in the form

$$y''_{r+1} + (1 + \beta_{0,r} + \alpha_{0,i})\, y_{r+1} = \phi + \beta_{0,r} y_r - H[y_r, y_i]. \tag{3.33}$$

Applying spectral collocation on equation (3.2.2) results in

$$\left(\mathbf{D}^2 + (1 + \beta_{0,r} + \alpha_{0,i})\mathbf{I} \right) Y_{r+1} = \phi + \beta_{0,r} y_r - H[y_r, y_i]. \tag{3.34}$$

This equation has the form

$$\mathbf{M} Y_{r+1} = \Phi_r. \tag{3.35}$$

where $\mathbf{D}$ is the scaled Chebyshev differentiation matrix, $\mathbf{M} = \mathbf{D}^2 + (1 + \beta_{0,r} + \alpha_{0,i})\mathbf{I}$ and $\mathbf{I}$ is the identity matrix. The solutions are presented in Tables 3.6 to 3.8 and Figure 3.3.

Table 3.6: Numerical results of equation (3.14) compared with the results in Hosen and Chowdhury[110].

x	ho-SQLM	Runge-Kutta method [110]	HBM [110]
0.0	0.174532	0.174532	0.174532
0.5	0.132217	0.132306	0.132217
1.0	0.026019	0.026059	0.026019
1.5	-0.092692	-0.092797	-0.092692
2.0	-0.166728	-0.166751	-0.166728
2.5	-0.159977	-0.160017	-0.159977
3.0	-0.075757	-0.075853	-0.075757
3.5	0.044947	0.045014	0.044946
4.0	0.144027	0.144100	0.144027
4.5	0.173455	0.173458	0.173455
5.0	0.118784	0.118883	0.118785

Table 3.6 shows the numerical results for the higher-order spectral quasilinearization method and some numerical results in the literature. The results are in good agreement with those obtained by Hosen and Chowdhury using the harmonic balance method (HBM). The results are slightly different to those they obtained through the Runge-Kutta method.

Table 3.7: Comparison of the convergence of the HO-SQLM and the SQLM for $-y''(0)$.

iter.	HO-SQLM	SQLM
1	0.35349768	0.35349769
2	0.35349768	0.35349772
3	0.35349768	0.35349767
4	0.35349768	0.35349768
5	0.35349768	0.35349768
6	0.35349768	0.35349768
7	0.35349768	0.35349768
8	0.35349768	0.35349768
9	0.35349768	0.35349767
10	0.35349768	0.35349767

Table 3.7 shows a comparison of the convergence of the spectral quasilinearization method and the higher-order spectral quasilinearization method. The SQLM converges after three iterations while the HO-SQLM converges at the first iteration. This shows that the HO-SQLM is a better method for this equation.

Table 3.8: Comparison of the residual error of the scheme for equation (3.14) using SQLM and HO-SQLM.

iter.	HO-SQLM	SQLM
1	4.93725122e-09	1.02214382e-02
2	4.18055163e-09	1.18952284e-05
3	1.25561667e-09	4.82083590e-09
4	7.65845937e-09	3.99137673e-09
5	2.98729458e-09	3.80220183e-09
6	3.23467714e-09	2.76901585e-09
7	4.34062269e-09	3.74399417e-09
8	2.11417966e-09	1.35748007e-09
9	6.78534445e-09	6.21781976e-09
10	4.32607078e-09	6.14506018e-09

Table 3.8 shows the residual errors for the equation (3.14) using the SQLM and the HO-SQLM. The residual error for SQLM is bigger than that for the HO-SQLM over the first three iterations. As the number of iterations increases the error improves for the SQLM. The error for the HO-SQLM is consistently good from the first iteration.

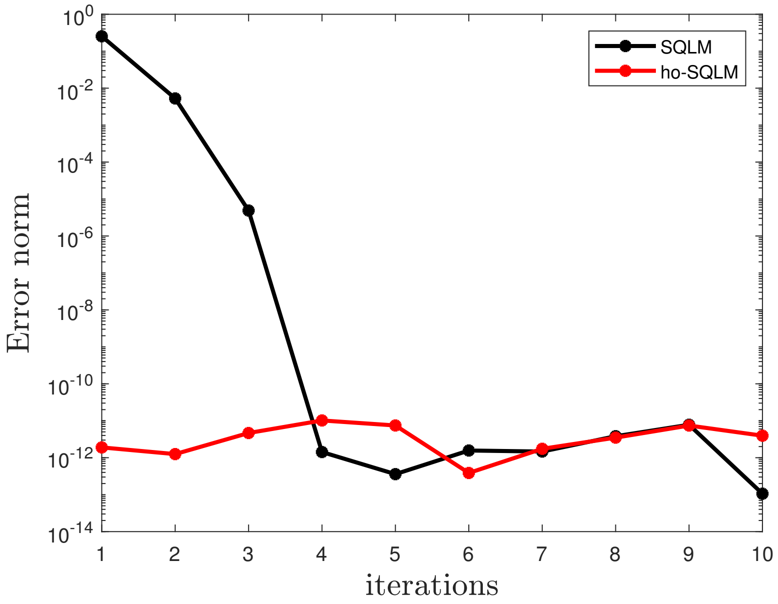


Figure 3.3: Convergence of the approximations $y(t)$ of equation (3.14) for $N = 80$.

Figure (3.3) shows the convergence graphs for the spectral quasilinearization and higher-order quasilinearization methods. The SQLM starts with a large error and improves as the number of iterations increases. The error from the HO-SQLM is small from the first iteration.

3.2.3 Example 3

Here, we consider [111]

$$y''(t) + y(t) - \lambda y(t)(1 - y'(t)^2) = 0, \qquad y(0) = 1, \quad y'(0) = 0, \tag{3.36}$$

we use the SQLM to linearize the equation Linearizing the equations using the SQLM scheme gives

$$a_{2,r}y''_{r+1} + a_{1,r}y'_{r+1} + a_{0,r}y_{r+1} = R_r, \tag{3.37}$$

with boundary conditions

$$\begin{aligned} y_{r+1} &= 0, \quad \text{at} \quad x = 1, \\ y'_{r+1} &= 0, \quad \text{at} \quad x = 0. \end{aligned} \tag{3.38}$$

Where

$$a_{2,r} = 1, \quad a_{1,r} = 2\lambda y_r y'_r, \quad a_{0,r} = 1 - \lambda + \lambda(y'_r)^2, \tag{3.39}$$

$$R_r = 2\lambda y_r y'_r. \tag{3.40}$$

The solution can be presented in the form

$$A\mathbf{F} = \boldsymbol{R_r}, \tag{3.41}$$

where

$$A = \mathbf{D}^2 + a_{1,r}\mathbf{D} + a_{0,r}\mathbf{I}. \tag{3.42}$$

The boundary conditions (3.38) are

$$y_{r+1}(z_N) = 1, \tag{3.43}$$

$$y'_{r+1}(z_N) = 0. \tag{3.44}$$

When applying the HO-SQLM, the nonlinear component is written as

$$N = \lambda y y'^2, \tag{3.45}$$

and approximated as

$$N \approx \lambda y_i y_i'^2 + \alpha_{0,i}(y - y_i) + \alpha_{1,i}(y' - y_i') + H[y, y', y_i, y_i'], \tag{3.46}$$

where

$$\alpha_{0,i} = \lambda y_i'^2, \qquad \alpha_{1,i} = 2\lambda y_i y_i'. \tag{3.47}$$

Equation (3.36) is now written as

$$y'' + (1 - \lambda)y + \lambda y_i y_i'^2 + \alpha_{0,i}(y - y_i) + \alpha_{1,i}(y' - y_i') + H[y, y', y_i, y_i'] = 0. \tag{3.48}$$

This equation becomes

$$y'' + \alpha_{1,i}y' + (1 - \lambda + \alpha_{0,i})y + H[y, y', y_i, y_i'] = \phi, \tag{3.49}$$

where

$$\phi = \alpha_{0,i}y_i + \alpha_{1,i}y_i' - \lambda y_i y_i'^2, \tag{3.50}$$

and

$$H[y, y', y_i, y_i'] = \lambda y y' 2 - \lambda y_i y_i'^2 - \alpha_{0,i}(y - y_i) - \alpha_{1,i}(y' - y_i'). \tag{3.51}$$

We now write $H[y, y', y_i, y_i']$ as

$$H[y, y', y_i, y_i'] = H[y_r, y_r', y_i, y_i'] + \beta_{0,r}(y_{r+1}, y_r) + \beta_{1,r}(y_{r+1}', y_r'), \tag{3.52}$$

where

$$\beta_{0,r} = \lambda y_r'^2 - \alpha_{0,i}, \qquad \beta_{1,r} = 2\lambda y_r y_r' - \alpha_{1,i}. \tag{3.53}$$

The numerical scheme for equation (3.49), upon applying spectral collocation is given as

$$\left(\mathbf{D}^2 + (\alpha_{1,i} + \beta_{1,r})\mathbf{D} + (1 - \lambda + \alpha_{0,i} + \beta_{0,r})\mathbf{I}\right) Y_{r+1} = \phi + \beta_{0,r} y_r + \beta_{1,r} y_r' - H[y_r, y_r', y_i, y_i']. \tag{3.54}$$

This equation has the form

$$\mathbf{M} Y_{r+1} = \Phi_r, \tag{3.55}$$

where

$$\mathbf{M} = \mathbf{D}^2 + (\alpha_{1,i} + \beta_{1,r})\mathbf{D} + (1 - \lambda + \alpha_{0,i} + \beta_{0,r})\mathbf{I}, \tag{3.56}$$

and

$$\Phi_r = \phi + \beta_{0,r} y_r + \beta_{1,r} y_r' - H[y_r, y_r', y_i, y_i'], \tag{3.57}$$

and, as before, **D** is the transformed Chebyshev differentiation matrix and **I** is the identity matrix, both of size $(N + 1)$ by $(N + 1)$.

The solution is presented in Tables 3.9 and 3.10 and Figure 3.4.

Table 3.9: Numerical results of equation (3.36) in comparison with the results of Alquran and Al-Khaled[111] and MATLAB inbuilt routine ode45 solver for $\lambda = 0.1$.

x	HO-SQLM	Ref[111]	Ref[111]	ode45
0.00	1.00000000	1.00000000	1.00000000	1.00000000
0.25	0.97198072	0.97039447	0.97198047	0.96900755
0.50	0.88920312	0.88374859	0.88918750	0.87899686
0.75	0.75558427	0.74618137	0.75541797	0.73802260
1.00	0.57784441	0.56675787	0.57700000	0.55719150
1.25	0.36558877	0.35630944	0.36279297	0.34851910
1.50	0.13107439	0.12652305	0.12418750	0.12355532
1.75	-0.11150949	-0.11048277	-0.12489453	-0.10696207
2.00	-0.34722526	-0.34221415	-0.36800000	-0.33267864
2.25	-0.56177395	-0.55585637	-0.58614453	-0.54290519
2.50	-0.74270802	-0.73870619	-0.75781250	-0.72614995

Table 3.9 shows the results obtained by higher-order spectral quasilinearization method compared with results in the literature. The results are in agreement with the ones in literature to a certain number of decimal places.

Table 3.10: Residual error of the scheme for equation (3.36) in comparison with spectral quasilinearization method for $\lambda = 0.1$.

iter.	HO-SQLM	SQLM
1	6.55185431e-08	3.46409204e-02
2	9.11008101e-08	1.22463639e-03
3	1.12282578e-08	4.42902092e-08
4	1.27713429e-07	8.67352355e-08
5	8.31496436e-09	3.51923518e-08
6	7.12694600e-08	3.90515197e-08
7	2.55153282e-08	1.22227357e-07
8	4.19619028e-08	2.10769940e-08
9	3.87430191e-08	9.62754712e-09
10	4.68920916e-08	4.68920916e-08

Table 3.10 shows the residual error of the HO-SQLM and the SQLM. The residual error for the HO-SQLM is much smaller than the error of SQLM. That shows that the HO-SQLM is more accurate than the SQLM.

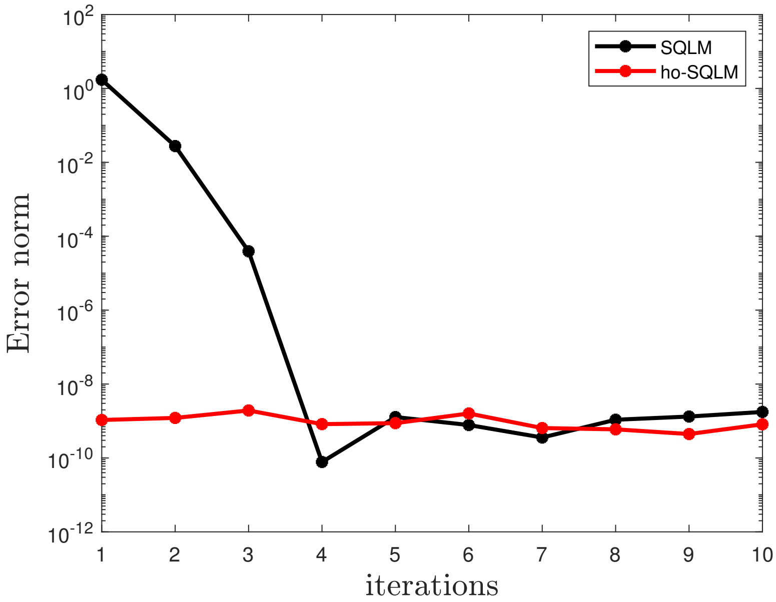


Figure 3.4: Convergence of the approximations $y(t)$ of equation (3.36) for $N = 80$.

Figure 3.4 shows the convergence graph of the SQLM and the HO-SQLM. The SQLM gives a bigger error over the first three iterations, but it reduces as the iterations increases. The HO-SQLM immediately gives a small error.

3.3 Summary

In this chapter, we have solved the nonlinear Bratu problem. Using the spectral quasilinearization method and the higher-order spectral quasilinearization method, we could compare the efficiency of the two methods. We chose the ho-SQLM over the SQLM since, with evidence, it is accurate. We included two examples that are initial value problems. Our tabular and graphical results show that, for this equation, the HO-SQLM is more suitable than the SQLM, since the evidence is that it is consistently the more accurate of the two. In chapter 4, we discuss and solve Troesch problem.

Chapter 4

The Troesch problem

In this chapter, we present a numerical solution of the highly nonlinear Troesch problem. The finite difference and higher-order spectral quasilinearization methods are used to solve the problem. A comparison is made between the two methods.

4.1 Literature review

Consider the nonlinear Troesch problem

$$y''(x) = n \sinh(ny(x)), \quad y(0) = 0, \quad y(1) = 1. \tag{4.1}$$

The equation has the exact solution

$$y(x) = \frac{4}{n} \tanh^{-1}(u(x)), \tag{4.2}$$

where n is a positive constant. The Troesch problem is a nonlinear ordinary differential equation, in the form of a second-order boundary value problem. It was first described by Weibel and Troesch [112]. The equation has been used in the study of confinement of a plasma column with radiation pressure [113]. Whereas Troesch solved the problem using the shooting method, other numerical methods have been employed. For example, Deeba et al. [114] solved the problem using the Adomian decomposition method. They found that the method gave accurate results for the Troesch problem. Chang [115] solved the Troesch problem successfully using the variational iteration method by transforming the hyperbolic nonlinear Troesch problem into a polynomial-type nonlinear problem. Although it was noted that the variational iterative method was used successfully to find the solution, when $n > 5$, the problem was still complex to solve. Mohyud-Din [116] used He's polynomials to solve the Troesch problem, and obtained the polynomials by a homotopy perturbation method. Roberts and Shipman [117] developed a closed-form solution for the Troesch problem and found a numerical solution for $n = 5$. Zarebnia and Sajjadian [118] applied the Sinc-Galerkin method to the Troesch problem. When they compared results from that method with the exact solutions and results from other numerical methods. They concluded that the method was applied successfully to the problem. To solve the Troesch problem, Feng et al. [119] presented a new algorithm which is a modification of the homotopy perturbation method. They compared the solutions obtained with those obtained by the variational iterative method and Adomian decomposition method. They found that their method gives accurate numerical solutions.

Bisheh-Niasar et al. [120] solved the Troesch problem by using new higher-order accurate schemes for solving two-point boundary value problems. They showed that the order of Scheme 1 was $O(h^4)$ and Scheme 2 was $O(h^6)$. They found that the methods were accurate. Khuri and Sayfy [121] presented a solution for the Troesch problem using the B-Spline collocation approach. Although they found that the method was efficient, they noted that the solution was less accurate as the value of n increased. Mohyud-Din [122] found the method of He's polynomial was effective in solving the Troesch problem. Ben-Romdhane and Temimi [123] solved the Troesch problem using a numerical technique that is based on the Newton-Raphson-Kantorovich scheme.In using the finite difference method to find the solution at each iteration, with $n = 500$ which is extremely large, they noted that the problem became too complex to solve. Doha et al. [124] successfully used the Jacobi-Gauss collocation method to find a solution for a Troesch problem. Makarov and Dragunov [125] tested a new technique, named the straight-inverse method, and showed it could be successfully applied to solve the Troesch problem. Their results showed good accuracy when compared to existing results in the literature. Chang [126] solved the Troesch problem using the shooting method. The results were in good agreement with those in the literature and the exact solution, so the method was judged to be efficient and reliable for solving the Troesch problem. Mirmoradi et al. [127] successfully applied the homotopy perturbation method to solve the Troesch problem, which they found to be accurate and efficient.

4.2 Solution

The solutions were obtained using the higher-order spectral quasilinearization method and the finite difference method. With these methods having been successfully applied to the Lane-Emden equation (see Chapter 2) and Bratu equation (see Chapter 3), the aim here was to show whether these methods could work for a problem as complicated as the Troesch problem. Now we show the solution for the higher-order spectral quasilinearization method. Consider equation (4.1), for which the nonlinear component is given as

$$N = -n \sinh{(ny)},$$

and approximated as

$$N \approx -\sinh{(ny_i)} + \alpha_{0,i}(y - y_i) + H[y, y_i],$$

where

$$H[y, y_i] = -n \sinh(ny_i) + n \sinh(y_i) - \alpha_{0,i}(y - y_i), \tag{4.3}$$

with

$$\alpha_{0,i} = -n^2 \cosh{(ny_i)}.$$

Equation (4.1) becomes

$$y''_{r+1} + (1 + \alpha_{0,i})\, y_{r+1} + H[y_r, y_i] + \beta_{0,r}\,(y_{r+1} - y_r) = \phi, \tag{4.4}$$

where

$$\phi = n\sinh(ny_i) + \alpha_{0,i}y_i, \tag{4.5}$$

and

$$\beta_{0r} = -n^2\cosh(ny_r) - \alpha_{0,i}. \tag{4.6}$$

This equation can be written in the form

$$\mathbf{M}Y_{r+1} = \Phi_r. \tag{4.7}$$

The finite difference discretization for equation (4.1) is

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - n\sinh\left(ny_i\right) = 0.$$

The equation becomes

$$y_{i+1} = h^2 n\sinh(ny_i) + 2y_i - y_{i-1} \tag{4.8}$$

The solution for the Troesch problem, according to the HO-SQLM, is obtained through iterations by considering the boundary conditions.

The solution is presented in Table 4.1, 4.2 and Figure 4.1. Results from the finite differences method are shown in Table 4.2.

Table 4.1: Exact solution and numerical solution by the HO-SQLM for the Troesch problem for $n = 0.5$.

x	Exact solution	HO-SQLM	Error
0.00	0.0000000000	0.0000000000	0.0000000000e+00
0.02	0.0234700578	0.0234700578	2.3942653415e-14
0.10	0.0916153363	0.0916153363	1.3045120539e-15
0.21	0.1980162262	0.1980162262	1.6486811916e-14
0.35	0.3329941556	0.3329941556	4.1633363423e-15
0.50	0.4845471647	0.4845471647	1.9428902931e-15
0.65	0.6390233518	0.6390233518	4.9960036108e-15
0.79	0.7816820569	0.7816820569	1.6209256160e-14
0.90	0.8976330073	0.8976330073	2.5535129566e-15
0.98	0.9735503452	0.9735503452	2.9976021665e-14
1.00	1.0000000000	1.0000000000	4.4408920985e-16

Table 4.1 shows the exact solution of the Troesch problem compared with solution and error from the higher-order spectral quasilinearization method. It is clear that the error is very small, which shows that the method is accurate.

Table 4.2: Solution of equation (2.7): exact solution, solution by finite difference method and results in the literature at different values of x

x	Exact	Finite difference	HPM[113]	ADM[113]
0.1	0.0959443493	0.0959460130	0.0959443155	0.0959383534
0.2	0.1921287477	0.1921319831	0.1921286848	0.1921180592
0.3	0.2887944009	0.2887990224	0.2887943176	0.2887803297
0.4	0.3861848464	0.3861905708	0.3861847539	0.3861687095
0.5	0.4845471647	0.4845536067	0.4845470753	0.4845302901
0.6	0.5841332484	0.5841399124	0.5841331729	0.5841169798
0.7	0.6852011483	0.6852074191	0.6852010943	0.6851868451
0.8	0.7880165227	0.7880216532	0.7880164925	0.7880055691
0.9	0.8928542161	0.8928573118	0.8928542059	0.8928480234

Table 4.2 shows a comparison between the exact solution for the Troesch problem, numerical solution by the finite difference method and other methods from literature, namely, the homotopy perturbation method (HPM) and the Adomian decomposition method (ADM). The results are all in agreement with the exact solution.

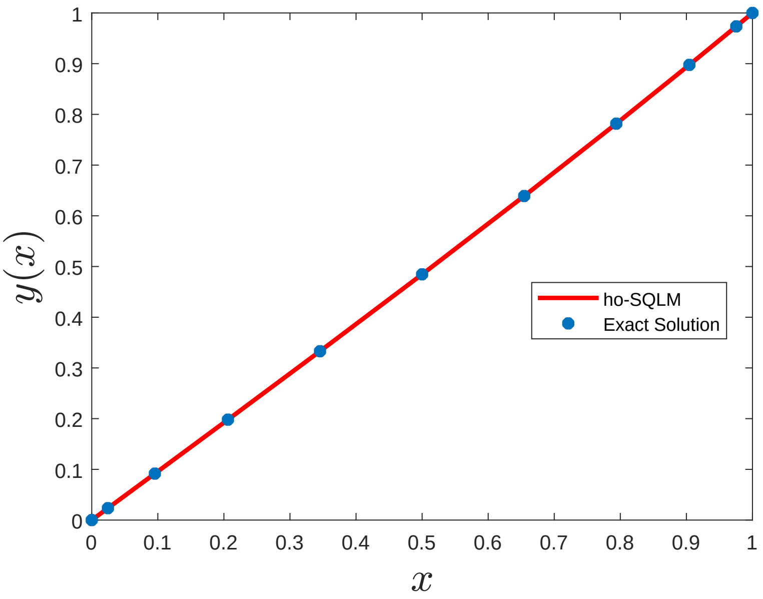


Figure 4.1: Comparison of the exact and the approximate solution for Troesch equation using the HO-SQLM.

Figure 4.1 shows a comparison between the exact solution and the approximate solution from the higher-order spectral quasilinearization method. The higher-order spectral quasilinearization method converges to the exact solution.

4.3 Summary

In this chapter, we showed the solution to the Troesch problem according to two numerical methods; the higher-order spectral quasilinearization method and the finite difference method. The higher-order spectral quasilinearization method performed better than the finite difference method.

Chapter 5

Conclusion

The aim of this study was to solve the Lane-Emden, the Bratu equations, and the Troesch problem using different numerical approaches. In this way we could investigate the use and effectiveness of different numerical methods. The Lane-Emden equation was solved using the multi-domain spectral quasilinearization method and the finite difference method, while the Bratu problem was solved using the spectral quasilinearization method and the higher-order spectral quasilinearization method. The Troesch problem was solved using the higher-order spectral quasilinearization method and the finite difference method.

Chapter 1 was a review of past studies and an outline of relevant numerical methods. The methods included the finite difference method, finite element method, Adomian decomposition method, and spectral methods. In the methods that were used in this project, the relevant application was also shown.

In Chapter 2, a review of the literature was presented for the Lane-Emden equation. The nonlinear Lane-Emden equation was first solved using the finite difference method and the multi-domain spectral quasilinearization method. Then two other variants of the Lane-Emden equation introduced greater complexity.

These were solved using the same two methods. Accurate results were obtained for both methods over all three problems. The errors were obtained by using the exact solution. A comparison was made for the errors to determine the method that is more accurate. The multi-domain spectral quasilinearization method had the smaller error, which made it to be the more accurate method.

In Chapter 3, a literature review concerning the Bratu problem was presented. The Bratu equation was then solved using the spectral quasilinearization method and the higher-order spectral quasilinearization method. Three examples were presented, which were initial value problems. The numerical approximations, convergence, and residual errors were presented in tables and figures. The higher-order spectral quasilinearization method showed excellent results. The comparison between results from the two methods led us to conclude that the higher-order spectral quasilinearization method is more accurate than spectral quasilinearization method.

In Chapter 4, the Troesch problem was introduced and solved using various methods. The aim is to determine if the methods could solve such a complex problem. We use the higher-order spectral quasilinearization method and the finite difference method. The solutions were represented in tables and figures. The higher-order spectral quasilinearization method produced accurate results for the problem. The finite difference also produced good results in comparison to solutions from the literature.

This project has addressed the objectives given in the first chapter. In the comparison between the finite difference method and the multi-domain spectral quasilinearization method for the Lane-Emden equation, we prefer the multi-domain spectral quasilinearization method as a suitable method to use due to its better convergence, consistency, and stability. For the Bratu problem a comparison of the spectral quasilinearization method and the higher-order spectral quasilinearization method showed that the modified method performed better than did the original method. Accuracy and convergence were more clearly demonstrated for the higher-order spectral quasilinearization method. We also solved the Troesch problem using the higher-order spectral quasilinearization method and the finite difference method. We found that the higher-order spectral quasilinearization method performed better in finding the solution for the problem.

References

[1] V. Thomee, From finite differences to finite elements: A short history of numerical analysis of partial differential equations, Journal of Computational and Applied Mathematics, 128:1-54, 2001.

[2] J. D. Anderson, Computational fluid dynamics, Mc-Graw-Hill, New York, 1995.

[3] R.I. Hickson, S.I. Barry, G.N. Mercer, H.S. Sidhu, Finite difference schemes for multilayer diffusion, Mathematical and Computer Modelling, 54:210–220, 2011.

[4] Y.S. Wong, G. Li, Exact finite difference schemes for solving Helmholtz equation at any wave number, International Journal of Numerical Analysis and Modeling, 2(1):91-108, 2001.

[5] Moczo, P., J. Kristek, and L. Halada, The finite-difference method for seismologists, an introduction, Comenius University, Bratislava, 2004.

[6] S. Wang, Y. Lin, A finite-difference solution to an inverse problem for determining a control function in a parabolic partial differential equation, Inverse Problems, 5:631-640, 1989.

[7] M. Cui, Compact finite difference method for the fractional diffusion equation, Journal of Computational Physics, 228:7792–7804, 2009.

[8] M. Tayakout, B. Bernauer, Y. Toure, J. Sanchez, Modelling and simulation of a catalytic membrane reactor, Simulation Practice and Theory, 2:205-219, 1995.

[9] S.B. Yuste, L. Acedo, An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations, SIAM Journal on Numerical Analysis, 42(5):1862-1874, 2005.

[10] C. Chen, F. Liu, K. Burrage, Finite difference methods and a Fourier analysis for the fractional reaction–subdiffusion equation, Applied Mathematics and Computation 198:754–769, 2008.

[11] P. Zhuang, F. Liu, V. Anh, I. Turner, New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation. SIAM Journal on Numerical Analysis, 46(2):1079-1095, 2008.

[12] L.A. Martino, A. Osella, C. Dorso, J.L. Lanata, Fisher equation for anisotropic diffusion: Simulating South American human dispersals, Physical Review E, 76(3), p.031923.

[13] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, Bulletin of the American Mathematical Society, 49:1-23, 1943.

[14] O. C. Zienkiewicz, The finite element method in structural and continuum mechanics, McGraw-Hill, New York, 1971.

[15] J.N. Reddy, An introduction to nonlinear finite element method, Oxford University Press, New York, 2004.

[16] J.N. Reddy, An introduction to finite element analysis, McGraw-Hill, New York, 1993.

[17] J.M. Augustine, An Application of Modern Analytical Solution Techniques to Non-linear Partial Differential Equations,Unpublished MSc dissertation, University of KwaZulu-Natal, Durban, South Africa, 2013.

[18] T. Hughes, The finite element method, linear static and dynamic finite element analysis, Dover Publications, New York, 1987.

[19] S. Alrabeei, H.S. Musthafa, Finite element method for 2d and 3d linear elasto-stastic model, 2019.

[20] K. Hiltunen, A stabilized finite element method for particulate two-phase flow equations Laminar isothermal flow, Computer Methods in Applied Mechanics and Engineering, 147(3-4):387-399, 1997.

[21] C.I. Goldstein, A finite element method for solving Helmholtz type equations in waveguides and other unbounded domains, Mathematics of Computation, 39: 309-324, 1982.

[22] E. Marchandise, J. Remacle, A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows, Journal of Computational Physics, 219:780–800, 2006.

[23] S. Nagrath, K. E. Jansen, R.T. Lahey Jr., Computation of incompressible bubble dynamics with a stabilized finite element level set method, Computer Methods in Applied Mechanics and Engineering, 94:4565–4587, 2005.

[24] T.E. Tezduyar, Finite element methods for flow problems with moving boundaries and interfaces, Archives of Computational Methods in Engineering, 8:83–130, 2001.

[25] M. Boncut, Finite element approximation of the Navier-Stokes Equation, General Mathematics, 12(2):61–68, 2004.

[26] S.R. Chowdhury, R Narasimhan, A cohesive finite element formulation for modelling fracture and delamination in solids, Sadhana, 25(6):561–587, 2000.

[27] R. E. Bellman, R. Kalaba, Quasilinearization and Nonlinear Boundary-Value Problems, Rand Corporation, Santa Monica, 1965.

[28] P. G. Dlamini, S. S. Motsa, and M. Khumalo, On the comparison between compact finite difference and pseudospectral approaches for solving similarity boundary layer problems, Mathematical Problems in Engineering, Volume 2013, Article ID 746489, 15 pages, 2013.

[29] V.B. Mandelzweig, F. Tabakin, Quasilinearization approach to nonlinear problems in physics with application to nonlinear odes, Computer Physics Communications, 141:268–281, 2001.

[30] M. Koleva, L. Vulkov, Quasilinearization numerical scheme for fully nonlinear parabolic problems with applications in models of mathematical finance, Mathematical and Computer Modelling, 57:2564–2575, 2013.

[31] L. Wang, M Pei, W. Ge, A generalized quasilinearization method for nonlinear second-order impulsive differential equations involving the p-Laplacian, Acta Applicandae Mathematicae, 110: 247–257, 2010.

[32] A. Yakar, M.E. Koksal, Quasilinearization method for nonlinear differential equations with causal operators, Journal of Nonlinear Sciences and Application, 9: 1356–1364, 2016.

[33] V.A. Vijesha, R. Roya, G. Chandhini, A modified quasilinearization method for fractional differential equations and its applications, Applied Mathematics and Computation, 266:687-697, 2015.

[34] C. I. Gheorghiu,. Spectral methods for differential problems, Institute of Numerical Analysis, Cluj-Napoca, 2007.

[35] E.S. Maciel, Spectral method applied to unstructured thermochemical non-equilibrium re-entry flows in two dimensions: five species, Journal of Scientific and Engineering Research,5(2):100-129, 2018.

[36] M.Y. Hussaini, C.L. Streett, T.A. Zang, Spectral methods for partial differential equations, Transactions of the First Army Conference on Applied Mathematics and Computing, 1983.

[37] D Gottlieb, S. Gottlieb, Spectral methods for compressible reactive flows, Comptes Rendus Mécanique, 333:3–16, 2005.

[38] M. Pohl, F. Wenzel, M. Karrenbach, A Chebyshev method on 2D generalized coordinates, Wave Inversion Technology, 3:189-196.

[39] E.H. Doha, A.H. Bhrawy, S.S. Ezz-Eldien, Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations, Applied Mathematical Modelling, 35:5662-5672, 2011.

[40] D. R. Gardner, S. A. Trogdon, R. W. Douglass, A modified tau spectral method that eliminates spurious eigenvalues, Journal of Computational Physics, 80(1), 137–167, 1989.

[41] F.Ghoreishi, S.Yazdani, An extension of the spectral tau method for numerical solution of multi-order fractional differential equations with convergence analysis, Computers and Mathematics with Applications, 61:30-43, 2011.

[42] A. H. Bhrawy, M. A. Alghamdi, A Legendre tau-spectral method for solving time-fractional heat equation with nonlocal conditions, Scientific World Journal, Volume 2014, Article ID 706296, 7 pages, 2014.

[43] J. Shen, Efficient Spectral-Galerkin Method I. Direct solvers for the second and fourth-order Legendre equations using polynomials, SIAM Journal on Scientific Computing, 15(6):1489–1505, 1994.

[44] H. Muzara, S. Shateyi, G.T. Marewo, Spectral quasi-linearization method for solving the Bratu problem, Advances and Applications in Fluid Mechanics, 21(4):449-463, 2018.

[45] S.S. Motsa, A new spectral local linearization method for nonlinear boundary layer flow problems, Journal of Applied Mathematics, Volume 2013, Article ID 423628, 15 pages, 2013.

[46] C. RamReddy, T. Pradeepa, D. Srinivasacharya, Numerical study of mixed convection flow of a micropolar fluid towards permeable vertical plate with convective boundary condition, Journal of Applied Analysis and Computation, 6(2):254-270, 2015.

[47] R.A. Alharbey, H. Mondal, R. Behl, Spectral quasi-linearization method for non-darcy porous medium with convective boundary condition, Entropy, 21(9):838, 2019.

[48] H. Mondal, P. Sibanda, Spectral quasi-linearization method for entropy generation using the Cattaneo-Christov heat flux model. International Journal of Computational Methods, Volume 16, No. 3, Article ID 1940002, 18 pages, 2018.

[49] S.S. Motsa, P. G. Dlamini, M. Khumalo, Spectral relaxation method, and spectral quasilinearization method for solving unsteady boundary layer flow problems. Advances in Mathematical Physics, Volume 2014, Article ID 341964, 12 pages, 2014.

[50] T.M. Agbaje, S. Mondal, S.S. Motsa, P. Sibanda, A numerical study of unsteady non-Newtonian Powell-Eyring nanofluid over a shrinking sheet with heat generation and thermal radiation, Alexandria Engineering Journal, 56(1): 81-91, 2017.

[51] V.M. Magagula, S.S. Motsa, P. Sibanda, A multi-domain bivariate pseudospectral method for evolution equations, International Journal of Computational Methods, Volume 14, No. 2, Article ID 1750041, 27 pages, 2017.

[52] S.P. Goqo, S. Mondal, P. Sibanda, S.S. Motsa, Efficient multi-domain bivariate spectral collocation solution for MHD laminar natural convection flow from a vertical permeable flat plate with uniform surface temperature and thermal radiation, International Journal of Computational Methods, 16(06):1840029, 2019.

[53] M.S. Ayano, O. Otegbeye, S.P. Goqo, Natural convection MHD radiative flow on a sphere through porous medium considering Ohmic dissipation, International Journal of Heat and Technology, 37(4):999-1008, 2019.

[54] I.S. Oyelakin, S. Mondal, P. Sibanda, A multi-domain spectral method for non-Darcian mixed convection flow in a power-law fluid with viscous dissipation, Physics and Chemistry of Liquids, 56(6):771-789, 2018.

[55] I.S. Oleyakin, S. Mondal, P. Sibanda, S.S. Motsa, A multi-domain bivariate approach for mixed convection in a Casson nanofluid with heat generation, Walailak Journal of Science and Technology, 16(9): 681-699, 2019.

[56] S.S. Motsa, P. Sibanda, On extending the quasilinearization method to higher-order convergent hybrid schemes using the spectral homotopy analysis method, Journal of Applied Mathematics, Volume 2013, Article ID 879195, 9 pages, 2013.

[57] S.S. Motsa, O.D. Makinde, S. Shateyi, On new high order quasilinearization approaches to the nonlinear model of catalytic reaction in a flat particle, Advances in Mathematical Physics, Volume 2013, Article ID 350810, 2013.

[58] C. Chun, Iterative methods improving Newton's method by the Decomposition method, Computers, and Mathematics with Applications, 50:1559-1568, 2005.

[59] N.S. Elgazery, Numerical solution for the Falkner–Skan equation, Chaos, Soliton and Fractals, 35(4):738–746, 2008.

[60] G. Adomian, A Review of the Decomposition Method in Applied Mathematics, Journal of Mathematical Analysis and Applications, 135:501-544, 1988.

[61] H. Jafari, V. Daftardar-Gejji, Revised Adomian decomposition method for solving a system of nonlinear equations, Applied Mathematics and Computation, 175:1–7, 2006.

[62] H. Jafari, V. Daftardar-Gejji, Solving linear and nonlinear fractional diffusion and wave equations by Adomian decomposition, Applied Mathematics, and Computation, 180:488–497, 2006.

[63] H.N.A. Ismail, K. Raslan, A.A. Abd Rabboh, Adomian decomposition method for Burgers–Huxley and Burgers–Fisher equations, Applied Mathematics and Computation, 159:291–301, 2004.

[64] D.B. Dhaigude, G.A. Birajdar, V.R. Nikam, Adomian decomposition method for fractional Benjamin-Boha-Mahony-Burger's equations, International Journal of Applied Mathematics and Mechanics, 8(12): 42-51, 2012.

[65] J. A. Sanchez Cano, Adomian decomposition method for a class of nonlinear problems, International Scholarly Research Network, Volume 2011, Article ID 709753, 10 pages, 2011.

[66] D.J. Evans, K.R. Raslan, The Adomian decomposition method for solving delay differential equation, International Journal of Computer Mathematics, 82(1):49–54, 2005.

[67] D. Kaya, The use of Adomian decomposition method for solving a specific nonlinear partial differential equations, Bulletin of the Belgian Mathematical Society - Simon Stevin, 9(3):343-349, 2002.

[68] J.H. He, A new approach to nonlinear partial differential equations, Communications in Nonlinear Science and Numerical Simulation, 2(4):230–235, 1997.

[69] G.C. Wu, Challenge in the variational iteration method – A new approach to identification of the Lagrange multipliers, Journal of King Saud University–Science, 25:175–178, 2013.

[70] J.H. He, Variational iteration method for autonomous ordinary differential systems, Applied Mathematics and Computations, 114(2–3):115–123, 2000.

[71] A.M. Wazwaz, The variational iterational method: A reliable analytical tool for solving the linear and nonlinear wave equations, Computers and Mathematics with Applications, 54:926–932, 2007.

[72] X. Shang, D. Han, Application of the variational iteration method for solving the nth-order integro-differential equations, Journal of Computational and Applied Mathematics, 234:1442–1447, 2010.

[73] T. A. Abassy, M. A. El-Tawil, H. El-Zoheiry, Modified variational iteration method for Boussinesq equation, Computers and Mathematics with Applications, 54:955–965, 2007.

[74] O. Olumuyiwa, On decoupled quasilinearization methods for solving systems of nonlinear boundary value problems, MSc dissertation, University of KwaZulu-Natal, Pietermaritzburg, 2014.

[75] B. Cushman-Roisin, J. Beckers, Introduction to geophysical fluid dynamics physical and numerical aspects, Academic Press, 2011.

[76] S.D. Oloniiju, On a numerical study of Ekman flow of non-Newtonian fluids, MSc dissertation, University of KwaZulu-Natal, Pietermaritzburg, 2017.

[77] R.D. Benguria, The Lane-Emden equation revisited, Contemporary Mathematics, 327:11-20, 2003.

[78] R. Emden, Gaskugeln, B.G. Teubner, Chicago, 1907.

[79] S. Mukherjee, B. Roy, P.K. Chaterjee, Solution of the Lane-Emden equation by differential transform method, International Journal of nonlinear science, 12: 478-484, 2011.

[80] C. Mohan, A.R. Al-Bayaty, Power-series solutions of the Lane-Emden equation, Astrophysics and Space Science, 73(1):227-239, 1980.

[81] H. Adibi, A.M. Rismani, On using a modified Legendre-spectral method for solving singular IVPs of Lane-Emden type, Computers and Mathematics with Applications, 60:2126-2130, 2010.

[82] B.M. Mirza, Approximate analytic solutions of the Lane-Emden equations for a self-gravitating isothermal gas sphere, Monthly Notices of the Royal Astronomical Society, 395(4):2288-2291, 2009.

[83] A. Taghavi and S. Pearce, A solution to the Lane–Emden equation in the theory of stellar structure utilizing the tau method, Mathematical Methods in the Applied Sciences, 36(10):1240–1247, 2013.

[84] R.A. Van Gorder, An elegant perturbation solution for the Lane–Emden equation of the second kind, New Astronomy, 16:65–67, 2010.

[85] S. S. Motsa, S. Shateyi, A successive linearization method approach to solve Lane-Emden type of equations, Mathematical Problems in Engineering, 2012(3), 2012. Mathematical Problems in Engineering, vol. 2012, Article ID 280702, 14 pages, 2012. https://doi.org/10.1155/2012/280702

[86] A.H. Bhrawy, A.S. Alofi, A Jacobi–Gauss collocation method for solving nonlinear Lane–Emden type equations, Communications in Nonlinear Science and Numerical Simulation, 17:62–70, 2012.

[87] K. Boubaker, R.A. Van Gorder, Application of the BPES to Lane–Emden equations governing polytropic and isothermal gas spheres, New Astronomy, 17:565–569, 2012.

[88] M. Dehghan, F. Shakeri, Approximate solution of a differential equation arising in astrophysics using the variational iteration method, New Astronomy, 13: 53–59, 2008.

[89] E.H. Doha, W.M. Abd-Elhameed, Y.H. Youssri, Second kind Chebyshev operational matrix algorithm for solving differential equations of Lane–Emden type, New Astronomy, 23-24:113–117, 2013.

[90] N. Kumar, R.K. Pandey, C. Cattani, Solution of the Lane-Emden Equation using the Bernstein operational matrix of integration, International Scholarly Research Notices, vol. 2011, Article ID 351747, 7 pages, 2011.

[91] S. Iqbal, A. Javed, Application of optimal homotopy asymptotic method for the analytic solution of singular Lane–Emden type equation, Applied Mathematics and Computation, 217:7753–7761, 2011.

[92] Z. Smarda, Y. Khan, An efficient computational approach to solving singular initial value problems for Lane–Emden type equations, Journal of Computational and Applied Mathematics, 290:65–73, 2015.

[93] A.M. Rismani, H. Monfared, Numerical solution of singular IVPs of Lane–Emden type using a modified Legendre-spectral method, Applied Mathematical Modelling, 36:4830–4836, 2012.

[94] R. Iacono, M. De Feline, Constructing analytic approximate solutions to the Lane-Emden equation, Physics Letters A, 379:1802-1807, 2015.

[95] A.S.V. Ravi Kanth, K. Aruna, He's variational iteration method for treating nonlinear singular boundary value problems, Computers and Mathematics with Applications, 60:821-829, 2010.

[96] S.C. Shiralashetti, S. Kumbinarasaiah, Hermite wavelets operational matrix of integration for the numerical solution of nonlinear singular initial value problems, Alexendria Engineering Journal, 57:2591-2600, 2018.

[97] J. Jacobsen, K. Schmitt, The Liouville–Bratu–Gelfand problem for radial operators, Journal of Differential Equations, 184:283–298, 2002.

[98] G. Bratu, Sur les équations intégrales non linéaires, Bulletin de la Société Mathématique de France, 42: 113-142, 1914.

[99] R. Buckmire, On exact and numerical solutions of the one-dimensional planar Bratu problem, Mathematical Preprint Archive, 7:82-99, 2003.

[100] A. Mohsen, A simple solution of the Bratu problem, Computers and Mathematics with Applications, 67(1):26-33, 2014.

[101] O. Ragb, L.F. Seddek, M.S. Matbuly, Iterative differential quadrature solutions for Bratu problem, Computers and Mathematics with Applications, 74:249–257, 2017.

[102] H. Caglar, N. Caglar, M. Ozer, A. Valarıstos, A.N. Anagnostopoulos, B-spline method for solving Bratu's problem, International Journal of Computer Mathematics, 87(8):1885–1891 (2010).

[103] H.N. Hassan, M.S. Semary, Analytic approximate solution for the Bratu's problem by optimal homotopy analysis method, Communications in Numerical Analysis, 2013:1-14, 2013.

[104] H. Temimi, M. Ben-Romdhane, An iterative finite difference for solving Bratu's problem, Journal of Computational and Applied Mathematics, 292:76-82, 2016.

[105] S. A. Odejide and Y. A. S. Aregbesola, A note on two dimensional Bratu problem, Kragujevac Journal of Mathematics, 29(29):49-56, 2006.

[106] F. Ghomanjani, S. Shateyi, A new approach for solving Bratu's problem. Demonstratio Mathematica, 52(1):336-346, 2019.

[107] S. Tomar, R.K. Pandey, An efficient iterative method for solving Bratu-type equations, Journal of Computational and Applied Mathematics, 357:71-84, 2019.

[108] S.G. Venkatesh, S.K. Ayyaswamy, S.R. Balachandar, The Legendre wavelet method for solving initial value problems of Bratu-type. Computers and Mathematics with Applications, 63(8):1287-1295, 2012.

[109] M. Saravi, M. Hermann, D. Kaiser, Solution of Bratu's equation by He's variational iteration method, American Journal of Computational and Applied Mathematics, 3(1):46-48, 2013.

[110] M.A. Hosen, M.S.H. Chowdhury, A new reliable analytical solution for strongly non-linear oscillator with cubic and harmonic restoring force, Results in Physics, 5:111-114, 2015.

[111] M. Alquran, K. Al-Khaled, Effective approximate methods for strongly non-linear differential equations with oscillations, Mathematical Sciences, 6(1):32, 2012

[112] H. Temimi, H. Kurkcu, An accurate asymptotic approximation and precise numerical solution of highly sensitive Troesch's problem, Applied Mathematics and Computations, 235:253-260, 2014.

[113] H. Vazquez-Leal, Y. Khan,G. Fernandez-Anaya, A. Herrera-May, A. Sarmiento-Reyes, U. Filobello-Nino, V.M. Jimenez-Fernández, D. Pereyra-Diaz, A general solution for Troesch's problem, Mathematical Problems in Engineering, vol. 2012, Article ID 208375, 14 pages, 2012.

[114] E. Deeba, S.A. Khuri, S. Xie, An algorithm for solving boundary value problems, Journal of Computational Physics, 159:125–138, 2000.

[115] S. Chang, A variational iteration method for solving Troesch's problem, Journal of Computational and Applied Mathematics, 234:3043–3047, 2010.

[116] S.T. Mohyud-Din, Solution of Troesch's problem using He's polynomials, Revista de la Union Matematica Argentina, 52(1): 143–148, 2011.

[117] S. M. Roberts and J. S. Shipman, On the closed-form solution of Troesch's problem, Journal of Computational Physics, 21 (3): 291–304, 1976.

[118] M. Zarebnia and M. Sajjadian, The sinc-Galerkin method for solving Troesch's problem, Mathematical and Computer Modelling, 56(9-10):218–228, 2012.

[119] X. Feng, L. Mei, G. He, An efficient algorithm for solving Troesch's problem, Applied Mathematics and Computation, 189:500–507, 2007.

[120] M. Bisheh−Niasar, A. Saadatmandi, M. Akrami−Arani, A new family of high-order difference schemes for the solution of second-order boundary value problems, Iranian Journal of Mathematical Chemistry, 9(3):187–199, 2018.

[121] S.A.Khuri, A.Sayfy, Troesch's problem: A B-spline collocation approach, Mathematical and Computer Modelling, 54:1907-1918, 2011.

[122] S.T. Mohyud-Din, Solution of Troesch's problem using He's polynomials, Revista de la Union Matematica Argentina, 52:143-148, 2011.

[123] M. Ben-Romdhane, H. Temimi, A novel computational method for solving Troesch's problem with high-sensitivity parameter, International Journal for Computational Methods in Engineering Science and Mechanics, 18(4-5):230-237, 2017.

[124] E.H. Doha, D. Baleanu, A.H. Bhrawi , R.M. Hafez, A Jacobi collocation method for Troesch's problem in plasma physics, Proceedings of the Romanian Academy - Series A: Mathematics, Physics, Technical Sciences, Information Science, 15(2): 130–138, 2014.

[125] V.L. Makarov, D.V. Dragunov, An efficient approach for solving stiff nonlinear boundary value problems, Journal of Computational and Applied Mathematics, 345: 452-470, 2019.

[126] S. Chang, Numerical solution of Troesch's problem by simple shooting method, Applied Mathematics and Computation 216:3303–3306, 2010.

[127] S.H. Mirmoradia, I. Hosseinpoura, S. Ghanbarpourb, A. Barari, Application of an approximate analytical method to nonlinear Troesch's problem, Applied Mathematical Sciences, 3(32):1579-1585, 2009.

Appendix A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code for y'' + (2/x)y' + y^5 = 0.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MD-SQLM
 clear
 N = 100;
 %clc
 [D,tau] = cheb(N);
  T = 2; % x in [0,T]

  p = 100; % p is the number of intervals
 xx  = linspace(0,T,p+1);

 %--------------------------------------------
 %   solution in first interval
 scale = (xx(2)-xx(1));
 x = scale*(tau+1)/2;
 D1 = (2/(scale))*D; D2 = D1^2;

 %Define the function g(x,y) = -2(2x^2 + 3)y

 g = @(x,y)(y.^5);
 alpha0 = 1; beta0 = 0; %initial conditions y(0) and y'(0)

 iterations = 20;
 %initial guesses
 fr = ones(N+1,1); %Chose initial guess that satisfies conditions at x = 0

 for it = 1:iterations
 %Solve for y
 A1 = D1(1:N,1:N);
 R1 = fr(1:N)- D1(1:N,N+1)*alpha0;
 soly = A1\R1;
 yr = [soly;alpha0];
 %----------------------------------------------------------
 %Solve for f
 A2 = D1(1:N,1:N) + diag(2./x(1:N));
 R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*beta0;
 solf = A2\R2;
  fr = [solf;beta0];
 end

  yN = yr(1); % y(x_1)
  fN = fr(1); % f(x_1)
yapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
fapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
yapprox(1) = yN;     %Value of y(x1)
fapprox(1) = fN;     %Value of f(x1)


%---------------------------------------------
 %   solution in the interval [x_{i-1},x_i]
```

```
for i = 2:p
    alphai = yr(1); betai = fr(1);
    x = (xx(i+1)-xx(i))*tau/2 + (xx(i+1)+xx(i))/2;
   D1 = (2/(xx(i+1)-xx(i)))*D;
for it = 1:iterations
%------------------------------------------------------------
%Solve for y
A1 = D1(1:N,1:N);
R1 = fr(1:N) - D1(1:N,N+1)*alphai;
soly = A1\R1;
yr = [soly;alphai];
%------------------------------------------------------------
%Solve for f
A2 = D1(1:N,1:N) + diag(2./x(1:N));
R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*betai;
solf = A2\R2;
 fr = [solf;betai];
end
    yN = yr(1); % y(x_i)
    fN = fr(1); % f(x_i)
 yapprox(i) = yN;
 fapprox(i) = fN;
end
yapprox = [1; yapprox];

yexact = (1 + (xx.^2)/3).^(-1/2);

for j = 1:10:p
 fprintf('%10.2f\t%10.11f\t%10.11f\t %10.4e\n',xx(j),yapprox(j),yexact(j), abs(yappr
end
```

```
%
 plot(xx,yexact,'r*',xx,yapprox,'k-')
 ylabel('$y(x)$','FontSize',18,'InterPreter','Latex')
 xlabel('x','FontSize',18,'InterPreter','Latex')
 legend('y-exact','Multi-domain')
    set(gca,'fontsize',14)
    str = {'N = 100'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [0.89]; % the distance of texts in y-axis
text(w ,s,str) % setup display

% Finite difference method
clear all
clc

N = 60;

a = 0; b = 2;
h = (b-a)/N;

x = a:h:b;

for i = 2:N
    u(1) = 1;
    u(2) = u(1);
    u(i+1) = ((2*x(i) + 2*(h))*u(i) - x(i)*(h^2)*u(i)^5 - x(i)*u(i-1))/(x(i) + 2*h);

end

U = 1./sqrt(1 + (x.^2)/3);
```

```
Error = abs(u-U);
for j = 1:N+1
    fprintf('%10.2f\t %10.11f\t %10.11f \t %10.5e \n',x(j),u(j),U(j),Error(j));
end
plot(x,u,'k-',x,U,'r*')

%plot(x,U,'r*',x,u,'-k')
ylabel('$y(x)$','FontSize',18,'InterPreter','Latex')
 xlabel('x','FontSize',18,'InterPreter','Latex')
 legend('Finite difference','y-exact')
    set(gca,'fontsize',14)
    str = {'N = 60'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [0.9]; % the distance of texts in y-axis
text(w ,s,str) % setup display


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code for y'' + 2/x y' + 4*(2*exp(y)+exp(y/2)) = 0.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%MD-SQLM

 clear
 N = 100;
 %clc
 [D,tau] = cheb(N);
  T = 2; % x in [0,T]

  p = 100; % p is the number of intervals
 xx  = linspace(0,T,p+1);
```

```
%-------------------------------------------
%   solution in first interval
scale = (xx(2)-xx(1));
x = scale*(tau+1)/2;
D1 = (2/(scale))*D; D2 = D1^2;

%Define the function g(x,y) = -2(2x^2 + 3)y

g = @(x,y)(4*(2*exp(y)+exp(y/2)));

alpha0 = 0; beta0 = 0; %initial conditions y(0) and y'(0)

iterations = 20;
%initial guesses
fr = ones(N+1,1); %Chose initial guess that satisfies conditions at x = 0

for it = 1:iterations
%Solve for y
A1 = D1(1:N,1:N);
R1 = fr(1:N)- D1(1:N,N+1)*alpha0;
soly = A1\R1;
yr = [soly;alpha0];
%---------------------------------------------------------
%Solve for f
A2 = D1(1:N,1:N) + diag(2./x(1:N));
R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*beta0;
solf = A2\R2;
 fr = [solf;beta0];
end
```

```
  yN = yr(1); % y(x_1)
  fN = fr(1); % f(x_1)
yapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
fapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
yapprox(1) = yN;     %Value of y(x1)
fapprox(1) = fN;     %Value of f(x1)


%--------------------------------------------
 %   solution in the interval [x_{i-1},x_i]
 for i = 2:p
     alphai = yr(1); betai = fr(1);
     x = (xx(i+1)-xx(i))*tau/2 + (xx(i+1)+xx(i))/2;
    D1 = (2/(xx(i+1)-xx(i)))*D;
 for it = 1:iterations
 %----------------------------------------------------------
 %Solve for y
 A1 = D1(1:N,1:N);
 R1 = fr(1:N) - D1(1:N,N+1)*alphai;
 soly = A1\R1;
 yr = [soly;alphai];
 %----------------------------------------------------------
 %Solve for f
 A2 = D1(1:N,1:N) + diag(2./x(1:N));
 R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*betai;
 solf = A2\R2;
  fr = [solf;betai];
 end
     yN = yr(1); % y(x_i)
     fN = fr(1); % f(x_i)
  yapprox(i) = yN;
  fapprox(i) = fN;
 end
 yapprox = [0; yapprox];
 yexact = -2*log(1+xx.^2);

 for j = 1:10:p
  fprintf('%10.2f\t%10.11f\t%10.11f\t %10.4e\n',xx(j),yapprox(j),yexact(j), abs(yappr
 end


%
 plot(xx,yexact,'r*',xx,yapprox,'k-')

 ylabel('$y(x)$','FontSize',18,'InterPreter','Latex')
 xlabel('x','FontSize',18,'InterPreter','Latex')
 legend('y-exact','Multi-domain')
    set(gca,'fontsize',14)
    str = {'N = 100'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [-0.9]; % the distance of texts in y-axis
text(w ,s,str) % setup display


% Finite difference method
clear all
clc

N = 60;

a = 0; b = 2;
h = (b-a)/N;
```

```
x = a:h:b;


for i = 2:N
    u(1) = 0;
    u(2) = u(1);
    u(i+1) = ((2*x(i) + 2*(h))*u(i) - 4*x(i)*(h^2)*(2*exp(u(i))+exp(u(i)/2)) - x(i)*u

end
U = -2*log(1+x.^2);
Error = abs(u-U);
for j = 1:N+1
    fprintf('%10.2f\t %10.11f\t %10.11f \t %10.5e \n',x(j),u(j),U(j),Error(j));
end



plot(x,u,'k-',x,U,'r*')


%plot(x,U,'r*',x,u,'-k')
ylabel('$y(x)$','FontSize',18,'InterPreter','Latex')
 xlabel('x','FontSize',18,'InterPreter','Latex')
 legend('Finite difference','y-exact')
    set(gca,'fontsize',14)
str = {'N = 60'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [-0.9]; % the distance of texts in y-axis
text(w ,s,str) % setup display

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code for y'' + (2/x) y'' -6*y - 4*y*lny = 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% MD-SQLM
 clear
 N = 100;
 %clc
 [D,tau] = cheb(N);
  T = 2; % x in [0,T]

  p = 100; % p is the number of intervals
 xx  = linspace(0,T,p+1);

 %---------------------------------------------
 %   solution in first interval
 scale = (xx(2)-xx(1));
 x = scale*(tau+1)/2;
 D1 = (2/(scale))*D; D2 = D1^2;

 %Define the function g(x,y) = -2(2x^2 + 3)y

 g = @(x,y)(-6*y-4*y.*log(y));

 alpha0 = 1; beta0 = 0; %initial conditions y(0) and y'(0)

 iterations = 20;
 %initial guesses
 fr = ones(N+1,1); %Chose initial guess that satisfies conditions at x = 0

 for it = 1:iterations
 %Solve for y
 A1 = D1(1:N,1:N);
 R1 = fr(1:N)- D1(1:N,N+1)*alpha0;
 soly = A1\R1;
 yr = [soly;alpha0];
 %----------------------------------------------------------
 %Solve for f
 A2 = D1(1:N,1:N) + diag(2./x(1:N));
 R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*beta0;
 solf = A2\R2;
  fr = [solf;beta0];
 end

  yN = yr(1); % y(x_1)
  fN = fr(1); % f(x_1)
yapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
fapprox = zeros(p,1);%Initialize to save solution at x_i (end of intervals)
yapprox(1) = yN;     %Value of y(x1)
fapprox(1) = fN;     %Value of f(x1)


%---------------------------------------------
 %   solution in the interval [x_{i-1},x_i]
 for i = 2:p
     alphai = yr(1); betai = fr(1);
     x = (xx(i+1)-xx(i))*tau/2 + (xx(i+1)+xx(i))/2;
    D1 = (2/(xx(i+1)-xx(i)))*D;
 for it = 1:iterations
 %----------------------------------------------------------
 %Solve for y
 A1 = D1(1:N,1:N);
 R1 = fr(1:N) - D1(1:N,N+1)*alphai;
 soly = A1\R1;
 yr = [soly;alphai];
 %----------------------------------------------------------
 %Solve for f
 A2 = D1(1:N,1:N) + diag(2./x(1:N));
 R2 = -g(x(1:N),yr(1:N)) - D1(1:N,N+1)*betai;
 solf = A2\R2;
  fr = [solf;betai];
 end
     yN = yr(1); % y(x_i)
     fN = fr(1); % f(x_i)
  yapprox(i) = yN;
  fapprox(i) = fN;
 end
   yapprox = [1; yapprox];
 yexact = exp(xx.^2);

  for j = 1:10:p
  fprintf('%10.2f\t%10.11f\t%10.11f\t %10.4e\n',xx(j),yapprox(j),yexact(j), abs(yappr
 end


%
 plot(xx,yexact,'r*',xx,yapprox,'k-')
 ylabel('$y(x)$','FontSize',18,'InterPreter','Latex')
 xlabel('x','FontSize',18,'InterPreter','Latex')
 legend('y-exact','Multi-domain')
    set(gca,'fontsize',14)
    str = {'N = 100'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [45]; % the distance of texts in y-axis
text(w ,s,str) % setup display
```

```
%finite difference method
clear all
clc

N = 100;


a = 0; b = 2;
h = (b-a)/N;

x = a:h:b;

for i = 2:N
    u(1) = 1;
    u(2) = u(1);
    u(i+1) = ((2*x(i) + 2*(h) + 6*(h^2)*x(i))*u(i) - x(i)*u(i-1) + 4*h^2*x(i)*u(i)*lo

end

U = exp(x.^2);
Error = abs(u-U);
for j = 1:N+1
    fprintf(’%10.2f\t %10.11f\t %10.11f \t %10.5e \n’,x(j),u(j),U(j),Error(j));
end



plot(x,u,’k-’,x,U,’r*’)

%plot(x,U,’r*’,x,u,’-k’)
ylabel(’$y(x)$’,’FontSize’,18,’InterPreter’,’Latex’)
 xlabel(’x’,’FontSize’,18,’InterPreter’,’Latex’)
 legend(’Finite difference’,’y-exact’)
```

```
    set(gca,'fontsize',14)
str = {'N = 100'}; % here put what ever you want
w = [1.6]; % the distance of texts in x-axis
s = [46]; % the distance of texts in y-axis
text(w ,s,str) % setup display


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Code for y'' + lambda*exp(y) = 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SQLM AND ho-SQLM
clc
clear all

N = 80;
[D,x] = cheb(N);

a = 1*10^(-10); b = 1;
Lx = b - a;

eta = Lx*(x+1)/2;

D1 = (2/Lx)*D; D2 = D1^2; D3 = D1^3;

I = eye(N+1,N+1);

eps = 3.513830719;

yr = eta.*(eta-1); %zeros(N+1,1);

iterations = 10;
```

```
%sqlm
for r = 1:iterations

    yprev = yr;



    yr1= D1*yr;
    yr2= D2*yr;

    a0r = eps*exp(yr);

    A = D2 + diag(a0r)*I;

    Ry = a0r.*yr - eps*exp(yr);

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 0;
    A(1,:) = 0; A(1,1) = 1; Ry(1) = 0;

    Solution = A\Ry;

    yr = Solution(1:N+1);
    yr1= D1*yr;
    yr2 = D2*yr;



    error_yr(r) = norm(yr - yprev,inf);

    resy(:,r) = max(abs(yr2 + eps*exp(yr)));

    resnormyr(r) = norm(resy(:,r),inf);
    %fprintf('%10.0f \t %10.8e \n',r, resnormyr(r))
end

yd = eta.*(eta-1);%zeros(N+1,1);
yd1 = D1*yd;
yi = yr;
yi1 = D1*yi;

for ii = 1:iterations

    yprevi = yi;

    yi1= D1*yi;
    yi2= D2*yi;

    a0d = eps*exp(yd);

    b0i = eps*exp(yi) - a0d;

    gi = eps*exp(yi) - eps*exp(yd) - a0d.*(yi - yd);
    phi = a0d.*yd - eps*exp(yd);



    A = D2 +  diag(a0d + b0i)*I;

    Ry = phi + b0i.*yi - gi;

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 0;
    A(1,:) = 0; A(1,1) = 1; Ry(1) = 0;

    Solution = A\Ry;
    yi = Solution(1:N+1);
    yi1= D1*yi;
    yi2 = D2*yi;

    %-------------------------------------------------------------------------
    %ERROR ANALYSIS
    %-----------------------------------------------------------------------

    error_yi(ii) = norm(yi - yprevi,inf);

    resy(:,ii) = max(abs(yi2 + eps*exp(yi)));

    resnormyi(ii) = norm(resy(:,ii),inf);

    fprintf('%10.0f \t %10.8e \n',ii, resnormyi(ii))
end

theta_c = 4.79871456;
% theta_c = 0.3172227274;
theta = sqrt(2*eps)*cosh(0.25*theta_c);
u_exact = -2*log((cosh((eta - 0.5).*(0.5*theta)))./(cosh(0.25*theta)));
% ur0 = ur;

plot(eta, yr, 'k', eta, yi, 'bd', eta, u_exact, 'r*','LineWidth',2.2)
xlabel('$\eta$','Interpreter','Latex','FontSize',20)
ylabel('$u$','Interpreter','Latex','Rotation',90,'FontSize',20)

legend('SQLM','ho-SQLM', 'Exact Solution');
set(gca,'fontsize',14)
    str = {'N = 80'}; % here put what ever you want
```

```
w = [0.6]; % the distance of texts in x-axis
s = [0.6]; % the distance of texts in y-axis
text(w ,s,str) % setup display


figure(1)
semilogy(1:iterations,error_yi,'k-*','LineWidth',2)
xlabel('$iterations$','Fontsize', 16, 'InterPreter', 'Latex')
ylabel('$Error~norm$','Fontsize', 16, 'InterPreter','Latex')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code for y'' + y + a*y^3 + b*sin(y) = 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SQLM and ho-SQLM

clc
clear all

N = 80;
[D,x] = cheb(N);

a = 0; b = 1;
Lx = b - a;

eta = Lx*(x+1)/2;

D1 = (2/Lx)*D; D2 = D1^2; D3 = D1^3;
```

```
I = eye(N+1,N+1);

A_0 = pi/18;
a_0 = 1;

xr = A_0*cos(eta);

iterations = 10;

b_0 = 1;

%sqlm
for r = 1:iterations

    xprev = xr;


    xr1= D1*xr;
    xr2= D2*xr;

    a1r = 3*a_0*xr.^2 + b_0*cos(xr);

    A = D2 + I + diag(a1r)*I;

    Rx = a1r.*xr - (a_0*xr.^3 + b_0*sin(xr)) ;

    A(N+1,:) = 0; A(N+1,N+1) = 1; Rx(N+1) = A_0;
    A(N,:) = D1(N+1,:); Rx(N) = 0;

    Solution = A\Rx;
    xr = Solution(1:N+1);
    xr1= D1*xr;
    xr2 = D2*xr;



    error_xr(r) = norm(xr - xprev,inf);


    resx(:,r) = max(abs(xr2 + xr + a_0*xr.^3 + b_0*sin(xr)));


    resnormx(r) = norm(resx(:,r),inf);
%     fprintf('%10.0f \t %10.8f \n',r, -xr2(N+1))
    fprintf('%10.0f \t %10.8e \n',r, resnormx(r))
end

xd = A_0*cos(eta);
xi = xr;
for i = 1:iterations


    xprevi = xi;


    a0d = 3*a_0*xd.^2 + b_0*cos(xd);
    b0i = 3*a_0*xi.^2 + b_0*cos(xi) - a0d;


    gi = (a_0*xi.^3 + b_0*sin(xi)) - (a_0*xd.^3 + b_0*sin(xd)) - a0d.*(xi - xd);
    phi = a0d.*xd - (a_0*xd.^3 + b_0*sin(xd));


    Ar = D2 + I + diag(a0d + b0i)*I;
    Rr = phi + b0i.*xi - gi; %(a0d + b0i).*xi - (xi + a_0*xi.^3 + b_0*sin(xi));


    Ar(N+1,:) = 0; Ar(N+1,N+1) = 1; Rr(N+1) = A_0;
    Ar(N,:) = D1(N+1,:); Rr(N) = 0;
    Solution = Ar\Rr;

    xi = Solution(1:N+1);
    xi1= D1*xi;
    xi2 = D2*xi;



    error_xi(i) = norm(xi - xprevi,inf);

    resxi(:,i) = (xi2 + xi + a_0*xi.^3 + b_0*sin(xi));

    resnormxi(i) = norm(resxi(:,i),inf);
%     fprintf('%10.0f \t %10.8f \n',i, -xi2(N+1))
    fprintf('%10.0f \t %10.8e \n',i, resnormxi(i))

end

figure(1)
semilogy(1:iterations,error_xr,'k-*',1:iterations,error_xi,'r-*','LineWidth',2)
xlabel('iterations','Fontsize', 16, 'InterPreter', 'Latex')
ylabel('Error~norm','Fontsize', 16, 'InterPreter','Latex')
legend('SQLM','ho-SQLM');



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Code for y'' + y - lambda*y(1 - (y')^2) = 0.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all
```

```
N = 80;
[D,x] = cheb(N);

a = 0; b = 2.5;
Lx = b - a;

eta = Lx*(x+1)/2;

D1 = (2/Lx)*D; D2 = D1^2; D3 = D1^3;

I = eye(N+1,N+1);

eps = 0.1;

yr = ones(N+1,1);

iterations = 10;


%sqlm
for r = 1:iterations

    yprev = yr;



    yr1= D1*yr;
    yr2= D2*yr;

    a0r = eps*yr1.^2;
    a1r = 2*eps*yr.*yr1;
    A = D2 + diag(a1r)*D1 + diag(1 - eps + a0r)*I;

    Ry = a0r.*yr + a1r.*yr1 - eps*yr.*(yr1.^2);

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 1;
    A(N,:) = D1(N+1,:); Ry(N) = 0;

    Solution = A\Ry;

    yr = Solution(1:N+1);
    yr1= D1*yr;
    yr2 = D2*yr;

    %------------------------------------------------------------------------
    %ERROR ANALYSIS
    %----------------------------------------------------------------------

    error_yr(r) = norm(yr - yprev,inf);

    resy(:,r) = max(abs(yr2 + yr - eps*yr + eps*yr.*(yr1.^2)));

    resnormyr(r) = norm(resy(:,r),inf);
%     fprintf('%10.0f \t %10.8f \n',r, -yr2(N+1))
%     fprintf('%10.0f \t %10.8e \n',r, resnormyr(r))
end
```

```
%ho-sqlm
yd = ones(N+1,1);
yd1 = D1*yd;
yi = yr;
yi1 = D1*yi;

for i = 1:iterations

    ypreví = yi;

    yi1= D1*yi;
    yi2= D2*yi;

    a0d = eps*yd1.^2;
    a1d = 2*eps*yd.*yd1;

    b0i = eps*yi1.^2 - a0d;
    b1i = 2*eps*yi.*yi1 - a1d;

    gi = eps*yi.*(yi1.^2) - eps*yd.*(yd1.^2) - a0d.*(yi - yd) - a1d.*(yi1 - yd1);
    phi = a0d.*yd + a1d.*yd1 - eps*yd.*(yd1.^2);



    A = D2 + diag(a1d + b1i)*D1 + diag(1 - eps + a0d + b0i)*I;

    Ry = phi + b0i.*yi + b1i.*yi1 - gi;

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 1;
    A(N,:) = D1(N+1,:); Ry(N) = 0;

    Solution = A\Ry;
    yi = Solution(1:N+1);
    yi1= D1*yi;
    yi2 = D2*yi;

    %-------------------------------------------------------------------------
    %ERROR ANALYSIS
    %-----------------------------------------------------------------------

    error_yi(i) = norm(yi - yprevi,inf);

    resy(:,i) = max(abs(yi2 + yi - eps*yi.*(1 - yi1.^2)));

    resnormyi(i) = norm(resy(:,i),inf);
%     fprintf(’%10.0f \t %10.8f \n’,i, -yi2(N+1))
%     fprintf(’%10.0f \t %10.8e \n’,i, resnormyi(i))
end

 figure(1)
semilogy(1:iterations,error_yr,’k-*’,1:iterations,error_yi,’r-*’,’LineWidth’,2)
xlabel(’iterations’,’Fontsize’, 16, ’InterPreter’, ’Latex’)
ylabel(’Error~norm’,’Fontsize’, 16, ’InterPreter’,’Latex’)
legend(’SQLM’,’ho-SQLM’);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code for y’’ = n*sinh(n*y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%ho-SQLM
clc
clear all
```

```
N = 30;
[D,x] = cheb(N);

a = 0; b = 1;
Lx = b - a;

eta = Lx*(x+1)/2;

D1 = (2/Lx)*D; D2 = D1^2; D3 = D1^3;

I = eye(N+1,N+1);

n = 0.5;

yr = eta;

iterations = 10;
%sqlm
for r = 1:iterations

    yprev = yr;



    yr1= D1*yr;
    yr2= D2*yr;

    a0r = -n^2*cosh(n*yr);

    A = D2 + diag(a0r)*I;
    Ry = a0r.*yr + n*sinh(n*yr);

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 0;
    A(1,:) = 0; A(1,1) = 1; Ry(1) = 1;

    Solution = A\Ry;

    yr = Solution(1:N+1);
    yr1= D1*yr;
    yr2 = D2*yr;


    error_yr(r) = norm(yr - yprev,inf);

%     resy(:,r) = max(abs(yr2 + eps*exp(yr)));

    resnormyr(r) = norm(yr2 - n*sinh(n*yr),inf);

%     fprintf('%10.0f \t %10.8e \n',r, resnormyr(r))
end

yd = eta;%zeros(N+1,1);
yd1 = D1*yd;
yi = yr;
yi1 = D1*yi;

for ii = 1:iterations

    yprevi = yi;

    yi1= D1*yi;
    yi2= D2*yi;

    a0d = -n^2*cosh(n*yd);

    b0i = -n^2*cosh(n*yi) - a0d;

    gi = -n*sinh(n*yd) + n*sinh(n*yd) - a0d.*(yi - yd);
    phi = a0d.*yd + n*sinh(n*yd);



    A = D2 +  diag(a0d + b0i)*I;

    Ry = phi + b0i.*yi - gi;

    A(N+1,:) = 0; A(N+1,N+1) = 1; Ry(N+1) = 0;
    A(1,:) = 0; A(1,1) = 1; Ry(1) = 1;

    Solution = A\Ry;

    yi = Solution(1:N+1);
    yi1= D1*yi;
    yi2 = D2*yi;



    error_yi(ii) = norm(yi - yprevi, inf);
    resnormyi(ii) = norm(yi2 - n*sinh(n*yi),inf);

%     error_yi(ii) = norm(yi - yprevi,inf);
%     resy(:,ii) = max(abs(yi2 + eps*exp(yi)));
%     resnormyi(ii) = norm(resy(:,ii),inf);
```

```
   %fprintf('%10.0f  %10.8e \t %10.8e \n',ii,resnormyr(ii),resnormyi(ii))


  % fprintf('%10.2f \t %10.20f \t  %10.20f \n',ii,yr1(N+1),yi1(N+1))



end

for j = 1:N+1
     fprintf('%10.2f \t %10.10f \t  %10.10f \n',eta(j),yr(j),yi(j))
end
figure(1)
semilogy(1:iterations,error_yr,'k-*',1:iterations,error_yi,'r->','LineWidth',2)
xlabel('$iterations$','Fontsize', 16, 'InterPreter', 'Latex')
ylabel('$Convergence~Error$','Fontsize', 16, 'InterPreter','Latex')
legend('SQLM','ho-SQLM')
    str = {'N = 10'}; % here put what ever you want
% w = [8]; % the distance of texts in x-axis
% s = [10^-5]; % the distance of texts in y-axis
% text(w ,s,str) % setup display



% Finite difference method
clc
clear all
N = 30;
a = 0;
b = 1;
```

```
h = (b - a)/N;
u0 = 0;
uL = 1;

x = a:h:b;

u = zeros(1,N+1);
steps =10;
u(1) = u0;
u(N+1) = uL;
n = 0.5;

for j = 1:steps
    u(2:N) = (1/3)*(u(3:N+1) + u(2:N) +u(1:N-1) - n*h^2*sinh(n*u(2:N)));
end
for m = 1:N+1
fprintf('%10.2f\t %10.8f\n', x(m), u(m))
end
```