# An Application of the Sinc-Collocation Method in Oceanography

A Thesis Submitted to the Committee on Graduate Studies

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in the Faculty of Arts and Science

TRENT UNIVERSITY

Peterborough, Ontario, Canada

Applied Modelling and

Quantitative Methods M.Sc. Graduate Program

September 2013

# Abstract

An Application of the Sinc-Collocation Method in Oceanography

Yasaman Mohseniahouei

In this thesis, we explore the application of the Sinc-Collocation method to an oceanography model. The model of interest describes a wind-driven current with depth-dependent eddy viscosity and is formulated in two different systems; a complex-velocity system and a real-value coupled system. In general, the Sinc-based methods excel over other traditional numerical methods due to their exponentially decaying errors, rapid convergence and handling problems in the presence of singularities at end-points. In addition, the Sinc-Collocation approach that we utilize exploits first derivative interpolation, whose integration is less sensitive to numerical errors. We present several model problems to demonstrate the accuracy, and stability of the method. We compare the approximate solutions determined by the Sinc-Collocation technique with exact solutions and also with those obtained by the Sinc-Galerkin approach in earlier studies. Our findings indicate that the method we utilized outperforms those used in past studies.

**Keywords :** Boundary Value Problems, Eddy Viscosity, Oceanography, Sinc Numerical Methods, Wind-Driven Currents

# Acknowledgments

I would like to express my unbounded gratitude to my supervisors, Dr. Kenzu Abdella, and Dr. Marco Pollanen, whose expertise, understanding and patience made this thesis possible. They believed in my capabilities and gave me the opportunity to study here at Trent University. I owe my current achievements and future successes to their generosity.

I would like to thank Dr. Bruce Cater for his presence on my supervisory committee and Dr. Serge D'Alessio from the University of Waterloo for acting as my external examiner.

I warmly thank all the faculty and staff in the AMOD program, but most especially Dr. Richard Hurley, Dr. Sabine McConnel, Ms. Gina Collins and Ms. Sarah Honey. Without their help I would not have been able to complete my Master's thesis.

I would also like to extend my thanks to the Office of Graduate Studies, Financial Aid and CUPE Local 3908. A student's life is full of financial hardships and their kindness went a long way to easing my financial burdens so that I may focus on my studies.

I wholeheartedly would thank my best friend, Ryan Kwan, for all of his unconditional help and support. His kindness and precious advice in the hard days I encountered was like that of a brother. Without his supports, I would never be able to finish this thesis.

I also would like to thank my sister, brothers, and most especially, my mother. She is incredibly patient, wise, supportive, and an unending source of love which sustained me throughout these last two years.

Last, but not least, I would like to give my thanks to my late father and sister, Zahra, who could not be here to see this day. I wish happiness and peace on their souls.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

- ADM - Adomian Decomposition Method
- BCs - Boundary Conditions
- BVPs - Boundary Value Problems
- DTM - Differential Transformation Method
- FDM - Finite Difference Method
- HAM - Homotopy Analysis Method
- HPM - Homotopy Perturbation Method
- IVPs - Initial Value Problems
- MSM - Multiple Shooting Method
- ODEs - Ordinary Differential Equations
- OHAM - Optimal Homotopy Asymptotic Method
- PDEs - Partial Differential Equations

- SSM - Simple Shooting Method

- TPBVPs - Two Point Boundary Value Problems

- VIM - Variational Iteration Method

- 3D - Three-Dimensional

# Glossary

- $x^*$ : northward axis
- $y^*$ : eastward axis
- $z^*$ : downward axis
- $\hat{x}^*$ : unit vector in the positive direction of $x^*$ -axis
- $\hat{y}^*$ : unit vector in the positive direction of $y^*$-axis
- $D_0$ : ocean depth
- $\rho$ : ocean mass density
- $\tau_w$ : magnitude of tangential surface wind stress
- $\chi$ : the angle between the positive $x^*$-axis and the wind direction
- $q^*$ : the horizental wind drift current
- $U^*$ : the current speed in direction of $x^*$-axis
- $V^*$ : the current speed in direction of $y^*$-axis

- $A_v^*(z^*)$ : the vertical eddy viscosity
- $f$ : Coriolis parameter at latitude $\theta$
- $\Omega$ : angular speed of rotation of the earth
- $k_f$ : linear slip bottom stress coefficient
- $A_0$ : near surface turbulent eddy viscosity
- $D_E$ : Ekman depth
- $U_0$ : natural velocity scale
- $\kappa$ : depth ratio
- $\sigma$ : bottom friction parameter
- $u$ : reduced current velocity in direction of $x^*$-axis
- $v$ : reduced current velocity in direction of $y^*$-axis
- $w$ : complex velocity

# Chapter 1

# Introduction

In many fields of study, modelling the governing phenomena leads to a specific set of differential equations, called boundary value problems (BVPs). In most cases, deriving analytical solutions of BVPs is extremely hard or completely impossible. Therefore, various numerical methods were developed to attack these problems. Some of the well-known numerical approximations to BVPs are finite-difference methods [52], finite-element methods [7, 12], boundary element methods [60], shooting methods [6], spline methods [58], and Sinc methods.

Sinc methods reduce the governing differential or integral equations to a system of algebraic equations, which makes the problems easier to solve. It is well-known that the Sinc-based methods are dominant over other numerical methods, especially in the presence of singularities and semi-infinite domains [66]. They are also characterized by exponentially decaying errors and rapid convergence [32]. Sinc-based methods have been applied to diverse scientific and engineering problems comprising heat conduction [43, 49], population growth [4], inverse problems [45, 65], astrophysics problems [18, 54], medical imaging [68], elastoplastic problems [1], and oceanography models[36, 71]. Very recently, the application of the Sinc-Collocation approach to the telegraph equation [28] and the second type of the Painlevé equations [63] has been studied.

In general, there are two equivalent but distinct Sinc approaches: Sinc-Galerkin and Sinc-Collocation. In earlier studies, it has been shown that the Sinc-Collocation approach is superior to the Sinc-Galerkin method due to its simple implementation and possible extensions to more general BVPs [8].

In the past century, hydrodynamic models and their numerical solutions have been receiving lots of attention. The first wind-driven current models were one-dimensional systems based on the work of Ekman [17]. Eventually two- and three-dimensional models were developed [26, 27]. To derive approximate solutions to 3D models, several numerical methods employing spectral methods [39], B-spline approach [14], Chebyshev and Legendre polynomials [15] and eigenfunction approach [13] were developed. Recently, Sinc-Galerkin approaches have been applied to a wind-driven current model [36, 71].

The intent of this thesis is to illustrate an application of the Sinc-Collocation technique to a steady state model of wind-driven currents with a depth-dependent eddy viscosity in coastal regions and semi-enclosed seas. The model is found in the work of Winter et al. [71]. They formulated the model as a complex-valued ordinary differential equation (ODE) and applied the original Sinc-Galerkin approach to solve it. Later, Koonprasert and Bowers [36], developed a block matrix formulation for the Sinc-Galerkin technique and applied it to the same model formulated as a coupled system of differential equations. Here we demonstrate the results by applying a Sinc-Collocation approach to both the complex system and real-value coupled system. Our goal is to show that the Sinc-Collocation method provides more accurate approximations than the methods previously applied.

Following the introduction, we introduce boundary value problems (BVPs) and some of their well-known analytical and numerical solutions, in section 2. Section 3 is devoted to a background on the Sinc methods along with fundamentals and implementations. In section 4, we briefly explain the formulation of the hydrodynamic model found in [71]. In Section 5, we provide extensions of a Sinc-Collocation approach developed by Abdella [2] to apply to both complex velocity and coupled systems. Section 6, portrays the results and comparisons we made between our results and those found in earlier studies. Finally, in section 7 we discuss concluding remarks and future works.

# Chapter 2

# Boundary Value Problems

In this chapter, we present an introduction to well-known boundary value problems (BVPs) arising in science and engineering. As well, we provide a summary of their analytical and numerical solutions discussed in past studies.

## 2.1 Ordinary Differential Equations

In many disciplines we face problems which require describing how quantities change. To fulfill this requirement we need a mathematical formulation, called ordinary differential equations (ODEs), to model the problems. This is not the end of the work since we still need to solve the models. Studies show that for a vast majority of ODEs, there is no way to find a closed-form solution. This fact led us to rely on numerical methods to approximate the solutions.

ODEs can always be reduced to sets of first-order differential equations. As an example consider the following linear second-order ODE

$$\frac{d^2f}{dx^2} + \alpha(x)\frac{df}{dx} = \beta(x). \tag{2.1}$$

We can rewrite (2.1) as two first-order equations

$$\frac{df}{dx} = g(x), \tag{2.2}$$
$$\frac{dg}{dx} = \beta(x) - \alpha(x)g(x). \tag{2.3}$$

where g(x) is a new variable.

Defining a new variable is not always as easy as this example. In some cases we also need to include other factors in the equation or some exponents of the independent variable to prevent increment of the round-off errors.

ODEs together with a set of additional constraints (boundary conditions) are called boundary value problem (BVPs). BVPs are classified regarding their equations characteristics and type of boundary conditions (BCs). To be precise, boundary conditions are algebraic equations involving the values of the unknown function, say f(x), and its derivatives. They may be as simple as the value of the function at a certain point or as complex as a set of nonlinear algebraic equations [56]. Boundary conditions are critical in determining which numerical method is feasible to attack the BVPs. Regarding their boundary conditions, BVPs are categorized into two major families: initial value problems (IVPs) and two-point boundary value problems (TPBVPs). We will briefly introduce IVPs and TPBVPs in following sections.

## 2.2 Initial Value Problems

Indeed, IVPs are those BVPs whose boundary conditions are set at the initial-point of the domain on which the problem is defined. To keep consistency in our writing, here we consider a linear second-order IVP as an example. The more general forms of IVPs together with related theorems and definitions are provided in detail in [9].

The linear second-order ODE

$$f'' = p(x)f' + q(x)f + r(x), \;\; a \leq x \leq b. \tag{2.4}$$

together with initial conditions

$$f(a) = \alpha_1, \quad f'(a) = \alpha_2.$$

is considered as an initial value problem.

There are some numerical methods in which the original BVP can be transformed to an IVP to approximate the solutions. One of these methods is the shooting method for linear BVPs which we introduce in section 2.5.2.

## 2.3 Two-Point Bondary Value Problems

TPBVPs are those ODEs that are required to satisfy boundary conditions at more than one point of the domain, usually at initial- and end-points of the domain on which the problem is defined. As mentioned earlier, selecting a proper numerical method to solve a BVP relies on its boundary conditions. There are four main types of boundary conditions corresponding to TPBVPs. Considering the linear second-order ODE given by

$$f'' = p(x)f' + q(x)f + r(x), \;\; a \leq x \leq b \tag{2.5}$$

we have

$$\text{Dirichlet BC :} \qquad f(a) = \gamma_1, \;\; f(b) = \gamma_2$$

$$\text{Neumann BC :} \qquad f'(a) = \gamma_1, \;\; f'(b) = \gamma_2$$

$$\text{Robin (or Mixed) BC :} \;\; \alpha_1 f(a) + \alpha_2 f'(a) = \gamma_1, \;\; \beta_1 f(b) + \beta_2 f'(b) = \gamma_2$$

$$\text{Periodic BC :} \qquad f(a) = f(b), \;\; f'(a) = f'(b)$$

## 2.4 A Short Survey of Solutions to TPBVPs

TPBVPs occur frequently in a wide range of disciplines such as applied mathematics, engineering, several branches of physics including gas dynamics, nuclear physics, chemical reaction, studies of atomic structures, astrophysics, and control and optimization theory.

A solution to a TPBVP is a solution to the differential equation which also satisfies the corresponding boundary conditions. In most cases obtaining analytical solutions to TPBVPs is a hard or impossible task. Therefore, we rely on numerical methods to approximate the solutions.

Some of the most common analytical methods for solving TPBVPs are; Adomian decomposition method (ADM) [3], variational iteration method (VIM) [25], homotopy analysis method (HAM) [41, 42], optimal homotopy asymptotic method (OHAM) [46], differential transformation method (DTM) [5, 10], and perturbation method [51].

Numerical solutions of TPBVPs fall into two major classes: direct methods and indirect methods. In the indirect methods, we reduce the higher order TPBVPs to the equivalent system of first-order equations and then apply any numerical method to approximate the solutions. By enlarging the system, indirect approaches call for a huge calculation which is one of their disadvantages. In contrast, the direct methods are those in which the original order of the equations is not changed.

Frequent occurrence and significance of TPBVPs inspired many researchers to develop some effective numerical solution procedures to them. Regardless of direct and indirect numerical approaches, there are two main classes of numerical methods for solving TPBVPs: The shooting method and relaxation methods. In the next section, we will specifically introduce these two methods. Regardless of earlier classifications, some of the well-known numerical solutions of TPBVPs are; finite difference methods [52], finite element methods [7, 12], boundary element methods [60], shooting methods [29, 53], spline approximation methods [58], the neural network approach [40] and spectral methods.

## 2.5 The Shooting Method

### 2.5.1 Background

The shooting method is an iterative approach with a trial-and-error nature. To be precise, in each iteration we guess the value of the slope at the initial point of the desired interval and continue to solve the differential equation until we arrive at the final point. The solution will not necessarily satisfy the boundary conditions, but it helps us to get close to the exact one. In fact, we must learn from each trial.

In general, two approaches of shooting methods are considered: simple shooting method (SSM) and multiple shooting method (MSM). A simple shooting algorithm for BVPs is proposed in [21, 34, 38, 64]. A modified version of SSM is developed in [30]. The application of the shooting method to nonlinear second-order BVPs is discussed in [23]. In [6], the shooting method is combined with Adomian decomposition. In [50, 57], the shooting technique is suggested to solve non-self-adjoint and self-adjoint singularly perturbed BVPs subject to different types of boundary conditions. The work in [24] is dedicated to the application of the SSM combined with the modified Newton method to fibre Raman amplifier design. A SSM coupled with an iterative method in [20] is applied to a certain class of singular TPBVPS arising in thermodynamics, electrostatics, physics, and statistics. A portrait of The MSM can be found in [34, 48]. Those interested in shooting method are referred to [69] in which both SSM and MSM are explored in great detail. Finally, a comparison between the shooting method and the finite difference method is outlined in [55]. Here we just listed some of the many studies using the shooting method. In the following sections, we explain the shooting method in brief.

### 2.5.2 The Shooting Method For linear BVPs

Consider the linear boundary value problem

$$y'' = p(x)y' + q(x)y + r(x), \;\; a \leq x \leq b, \;\; y(a) = \gamma_1, \;\; y(b) = \gamma_2 \tag{2.6}$$

The shooting method for linear BVPs relies on replacing the BVP by two initial value problems

$$y'' = p(x)y' + q(x)y + r(x), \;\; a \leq x \leq b, \;\; y(a) = \gamma_1, \;\; y'(a) = 0, \tag{2.7}$$

and

$$y'' = p(x)y' + q(x)y, \;\; a \leq x \leq b, \;\; y(a) = 0, \;\; y'(a) = 1. \tag{2.8}$$

Assume that $y_1(x)$ and $y_2(x)$ are the solutions to (2.7) and (2.8) respectively. Then let

$$y(x) = y_1(x) + m\, y_2(x). \tag{2.9}$$

where $m = \frac{\gamma_2 - y_1(b)}{y_2(b)}$ and $y_2(b) \neq 0$.

Regarding (2.9) we have

$$y'(x) = y_1'(x) + m\, y_2'(x),$$

and

$$y''(x) = y_1''(x) + m\, y_2''(x).$$

Hence

$$\begin{aligned} y''(x) &= p(x)y_1' + q(x)y_1 + r(x) + m(p(x)y_2' + q(x)y_2) \\ &= p(x)\,(y_1' + my_2') + q(x)\,(y_1 + my_2) + r(x) \\ &= p(x)y'(x) + q(x)y(x) + r(x). \end{aligned} \tag{2.10}$$

Similarly,

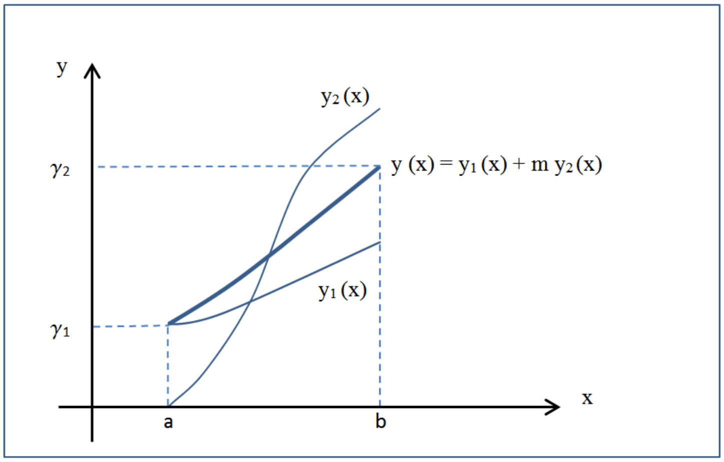
$$y(a) = y_1(a) + m\, y_2(a) = \gamma_1 + m\ .\ 0 = \gamma_1,$$

and

$$y(b) = y_1(b) + m\, y_2(b) = y_1(b) + \gamma_2 - y_1(b) = \gamma_2.$$

Therefore, we conclude that $y(x) = y_1(x) + m\, y_2(x)$ is the unique solution to (2.6). Various numerical methods such as the Euler's method, the Taylor

Figure 2.1: A schematic description of the linear shooting method



method, and the Runge-Kutta method exist to approximate the solutions $y_1(x)$ and $y_2(x)$. After obtaining these approximations, the solution to (2.6) will be approximated using equation (2.9). We illustrate the linear shooting method in Figure 2.1.

**Example 2.5.2.** (An application of the linear shooting method)

Here we examine the application of the linear shooting method to the following BVP found in reference [9].

Consider the boundary value problem

$$y''(x) = y'(x) + 2y(x) + \cos(x), \quad 0 \leq x \leq \pi/2,$$

Table 2.1: Errors of Example 2.5.2 (an application of the linear shooting method)

| $x_i$ | $w_i$ | $y(x_i)$ | $\vert w_i - y(x_i)\vert$ |
|---|---|---|---|
| 0. | -0.3 | -0.3 | |
| 0.16 | -0.311949732 | -0.311949948 | $2.17 \times 10^{-7}$ |
| 0.31 | -0.316218321 | -0.316218654 | $3.34 \times 10^{-7}$ |
| 0.47 | -0.312700639 | -0.312701007 | $3.67 \times 10^{-7}$ |
| 0.63 | -0.301483288 | -0.301483623 | $3.35 \times 10^{-7}$ |
| 0.79 | -0.282842457 | -0.282842712 | $2.55 \times 10^{-7}$ |
| 0.94 | -0.257237126 | -0.257237275 | $1.49 \times 10^{-7}$ |
| 1.10 | -0.225297762 | -0.225297802 | $3.97 \times 10^{-8}$ |
| 1.26 | -0.187810794 | -0.187810749 | $4.42 \times 10^{-8}$ |
| 1.41 | -0.145699244 | -0.145699173 | $7.05 \times 10^{-8}$ |
| 1.57 | -0.1 | -0.1 | |

with the boundary conditions

$$y(0) = -0.3, \;\; y(\pi/2) = -0.1.$$

and the exact solution $y(x) = -0.1(\sin(x) + 3\cos(x))$.

To solve this BVP, we utilized algorithm 11.1 in reference [9], page 649. We divided the interval [0, $\pi/2$] into 10 subintervals, resulting in 11 grid points, $x_i = a + ih$, where $h = (b-a)/10$, a=0 and b=$\pi/2$. The results are listed in Table 2.1. $w_i$ is an approximate solution for $y(x_i)$.

### 2.5.3 The Shooting Method For Nonlinear BVPs

As mentioned earlier, the shooting method is an alternate numerical technique to approximate solutions to BVPs. To illustrate the idea behind the shooting method, consider the general nonlinear, second-order TPBVP

$$\begin{aligned} g'' &= f(x, g, g'), \\ g(0) &= \gamma_1, \\ g(1) &= \gamma_2. \end{aligned} \tag{2.11}$$

In first step, the shooting method considers the initial value problem

$$\begin{aligned} g'' &= f(x, g, g'), \\ g(0) &= \gamma_1, \\ g'(0) &= \beta. \end{aligned} \tag{2.12}$$

As discussed in section 2.1, we transform higher-order BVPs as systems of two or more first-order equations. Therefore, we treat (2.12) as a system of two first-order equations

$$\begin{aligned} g' &\equiv v, \\ v' &= f(x, g, v), \\ g(0) &= \gamma_1, \\ v(0) &= \beta. \end{aligned} \tag{2.13}$$

Next step is solving (2.13) for different values of $\beta$, with the desire of finding a value of $\beta$ giving us a solution satisfying $g(1) = \gamma_2$. This is like trying different shooting angles while shooting a cannon from the point $(0,\gamma_1)$, aiming at the point (1,$\gamma_2$). Even though this method has a trial-and-error nature, we must try to learn from our misses. Indeed, the shooting method is trying to find the root of the nonlinear function

$$F(\beta) \equiv g_\beta(1) - \gamma_2. \tag{2.14}$$

where $g_\beta(x)$ is the solution of the initial value problem (2.12). In this step, the shooting method requires a root finding method. There are several root finding methods such as the bisection method, the Newton's method, the fixed-point iteration, and the secant method. An appropriate method to find the root of F($\beta$) might be the secant iteration which converges rapidly and requires no derivative of F($\beta$). To use the secant method, we need to guess initial approximations $\beta_0$ and $\beta_1$ and then generate subsequent approximations using

$$\beta_{n+1} = \beta_n - \frac{F(\beta_n)(\beta_n - \beta_{n-1})}{F(\beta_n) - F(\beta_{n-1})}. \tag{2.15}$$

To generate the sequence $\{\beta_i\}$, we may use the more powerful Newton's method. Here, the iteration is given by

$$\beta_{n+1} = \beta_n - \frac{F(\beta_n)}{F'_\beta(\beta_n)}. \tag{2.16}$$

To use the Newton's method, we only need to guess the initial approximation $\beta_0$.

A schematic description of the nonlinear shooting method is provided in

Figure 2.2: A schematic description of the nonlinear shooting method

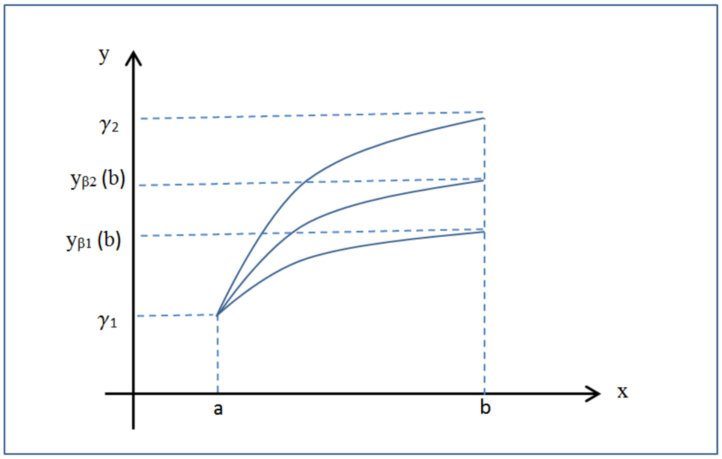


Figure 2.2.

## 2.6 Finite Difference Method

### 2.6.1 Background

To approximate the solution of differential equations, some numerical techniques rely on transforming the differential equations to a system of algebraic equations. The simplest way to generate such a system is by replacing the derivatives in the equation by finite difference formulae. This is the key idea behind finite difference methods. Even though finite difference approaches generally require more computation than the linear and nonlinear shooting methods, they exhibit better stability characteristics.

The term “finite differences” stems from this fact that the interpolation, differentiation, or integration of a function is approximated using differences

Table 2.2: Forward, backward and central differences

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $y_0$ | | | | $y_0$ | | | | $y_0$ | | | |
| | $\Delta y_0$ | | | | $\nabla y_1$ | | | | $\delta y_{\frac{1}{2}}$ | | |
| $y_1$ | | $\Delta^2 y_0$ | | $y_1$ | | $\nabla^2 y_2$ | | $y_1$ | | $\delta^2 y_1$ | |
| | $\Delta y_1$ | | $\Delta^3 y_0$ | | $\nabla y_2$ | | $\nabla^3 y_3$ | | $\delta y_{\frac{3}{2}}$ | | $\delta^3 y_{\frac{3}{2}}$ |
| $y_2$ | | $\Delta^2 y_1$ | | $y_2$ | | $\nabla^2 y_3$ | | $y_2$ | | $\delta^2 y_2$ | |
| | $\Delta y_2$ | | | | $\nabla y_3$ | | | | $\delta y_{\frac{5}{2}}$ | | |
| $y_3$ | | | | $y_3$ | | | | $y_3$ | | | |

based on values of that function at a finite number of points. There are three main notations for differences, called forward differences, backward differences and central differences, respectively denoted by the symbols $\Delta$, $\nabla$ and $\delta$. We illustrate these three notations in Table 2.2, assuming that we have four mesh points, $x_0 = a, x_1 = x_0 + h, x_2 = x_0 + 2h$ and $x_3 = x_0 + 3h = b$, where $h = \frac{(b-a)}{n}$. Here we have chosen n=3.

Indeed, the basic idea of any finite difference approach is inspired by the standard definition of the derivative of a function:

$$\text{right-handed approximation} \qquad y'_+(x) = \lim_{h \to 0} \frac{y(x+h) - y(x)}{h} \tag{2.17}$$

$$\text{left-handed approximation} \qquad y'_-(x) = \lim_{h \to 0} \frac{y(x) - y(x-h)}{h} \tag{2.18}$$

$$\text{centered approximation} \qquad y'_0(x) = \lim_{h \to 0} \frac{y(x+h) - y(x-h)}{2h} \tag{2.19}$$

As another advantage of the finite difference method, we can state the possibility of maintaining a specified order of truncation error by choosing a particular difference quotient and mesh size $h$.

### 2.6.2 Finite Difference Methods for Linear BVPs

This section is dedicated to a brief explanation of the finite difference method for linear BVPs. Consider the linear second-order BVP

$$f'' = p(x)f' + q(x)f + r(x), \;\; a \leq x \leq b, \;\; f(a) = \gamma_1, \;\; f(b) = \gamma_2. \tag{2.20}$$

The finite difference method replaces the $f'$ and $f''$ in (2.20) by difference-quotient approximations. To generate proper difference-quotients, we first need to select a positive integer $N$ and evenly divide the interval [a, b] into $(N + 1)$ subintervals. Endpoints of these subintervals can be obtained by $x_i = a + ih$, for i = 0, 1,..., N+1, where $h = \frac{(b-a)}{N+1}$.

At the interior mesh points, i.e. $x_i$ where i=1, 2,..., N, we have

$$f''(x_i) = p(x_i)f'(x_i) + q(x_i)f(x_i) + r(x_i), \tag{2.21}$$

Assuming the unknown function $f \in C^4[x_{i-1}, x_{i+1}]$, and approximating it by Taylor polynomial we have

$$f(x_{i+1}) = f(x_i + h) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(4)}({\xi_i}^{+}), \tag{2.22}$$

for some ${\xi_i}^+ \in (x_i, x_{i+1})$. Similarly,

$$f(x_{i-1}) = f(x_i - h) = f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(4)}({\xi_i}^-), \tag{2.23}$$

for some ${\xi_i}^- \in (x_i, x_{i+1})$. Adding (2.22) and (2.23) leads to

$$f(x_{i-1}) + f(x_{i+1}) = 2f(x_i) + h^2 f''(x_i) + \frac{h^4}{24}[f^{(4)}({\xi_i}^+) + f^{(4)}({\xi_i}^-)], \tag{2.24}$$

Solving (2.24) for $f''(x_i)$ gives

$$f''(x_i) = \frac{1}{h^2}[f(x_{i+1}) - 2f(x_i) + f(x_{i-1})] - \frac{h^2}{24}[f^{(4)}({\xi_i}^+) + f^{(4)}({\xi_i}^-)]. \tag{2.25}$$

Then we can utilize the intermediate value theorem to simplify (2.25) to

$$f''(x_i) = \frac{1}{h^2}[f(x_{i+1}) - 2f(x_i) + f(x_{i-1})] - \frac{h^2}{12}f^{(4)}(\xi_i). \tag{2.26}$$

Equation (2.26) is called the centered-difference approximation for $f''(x_i)$ with truncation error of $O(h^2)$. To obtain a centered-difference formula for $y'(x_i)$, we first need to subtract (2.23) from (2.22). The rest of the process is similar to what we have done to obtain (2.26), resulting in

$$f'(x_i) = \frac{1}{2h}[f(x_{i+1}) - f(x_{i-1})] - \frac{h^2}{6}f'''(\eta_i), \tag{2.27}$$

for some $\eta_i \in (x_{i-1}, x_{i+1})$.

Replacing $f'(x_i)$ and $f''(x_i)$ in (2.21) by centered-difference formulas (2.26) and (2.27) results in

$$\begin{aligned}\frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} = p(x_i)\left[\frac{f(x_{i+1}) - f(x_{i-1})}{2h}\right] &+ q(x_i)f(x_i) \\ +r(x_i) - \frac{h^2}{12}[2p(x_i)f'''(\eta_i) - f^{(4)(\xi_i)}]&.\end{aligned} \tag{2.28}$$

Now we rewrite (2.28) and boundary conditions $f(a) = \gamma_1$ and $f(b) = \gamma_2$ as

$$w_0 = \gamma_1, \qquad w_{N+1} = \gamma_2,$$

and

$$\left(\frac{-w_{i+1} + 2w_i - w_{i-1}}{h^2}\right) + p(x_i)\left(\frac{w_{i+1} - w_{i-1}}{2h}\right) + q(x_i)w_i = -r(x_i), \tag{2.29}$$

for i=1,2,...,N.

Equation (2.29) can be rearranged as

$$-\left(1 + \frac{h}{2}p(x_i)\right)w_{i-1} + (2 + h^2q(x_i))w_i - \left(1 - \frac{h}{2}p(x_i)\right)w_{i+1} = -h^2r(x_i). \tag{2.30}$$

The latter system of equations can be expressed in the tridiagonal $N \times N$ matrix form

$$\mathcal{A}W = \mathcal{B}, \tag{2.31}$$

where

$$\mathcal{A} = \begin{pmatrix} 2+h^2q(x_1) & -1+\frac{h}{2}p(x_1) & & & \\ -1-\frac{h}{2}p(x_2) & 2+h^2q(x_2) & -1+\frac{h}{2}p(x_2) & & \text{\Large 0} \\ & \ddots & \ddots & \ddots & \\ & \text{\Large 0} & & & -1+\frac{h}{2}p(x_{N-1}) \\ & & & -1-\frac{h}{2}p(x_N) & 2+h^2q(x_N) \end{pmatrix},$$

$$W = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{pmatrix},$$

and

$$\mathcal{B} = \begin{pmatrix} -h^2r(x_1) + \left(1+\frac{h}{2}p(x_1)\right)w_0 \\ -h^2r(x_2) \\ \vdots \\ -h^2r(x_{N-1}) \\ -h^2r(x_N) + \left(1-\frac{h}{2}p(x_N)\right)w_{N+1} \end{pmatrix}.$$

**Theorem 2.6.1**

Suppose that p(x), q(x), and r(x) are continuous on [a, b]. If $q(x) \geq 0$ on the interval [a, b], then the tridiagonal linear system (2.31), has a unique solution provided that $h < 2/L$, where $L = max_{a\leq x\leq b}|p(x)|$.

**Example 2.6.2**. (An application of the linear finite difference method) Here we examine the application of the linear finite difference method to the same BVP discussed in Example 2.5.2.

Consider the boundary value problem

$$y''(x) = y'(x) + 2y(x) + \cos(x), \quad 0 \leq x \leq \pi/2,$$

with the boundary conditions

$$y(0) = -0.3, \;\; y(\pi/2) = -0.1.$$

and the exact solution $y(x) = -0.1[\sin(x) + 3\cos(x)]$.

To solve this BVP, we utilize the algorithm 11.3 in reference [9], page 662. We divide the interval [0, $\pi/2$] into 10 subintervals, resulting in 11 grid points, $x_i = a + ih$, where $h = (b - a)/10$, a=0, and b=$\pi$/2. The results are depicted in Table 2.3. $w_i$ is an approximate solution for $y(x_i)$.

### 2.6.3 Finite Difference Methods for Nonlinear BVPs

Consider the general nonlinear BVP

$$y'' = f(x, y, y'), \;\; a \leq x \leq b, \;\; y(a) = \gamma_1, \;\; y(b) = \gamma_2. \tag{2.32}$$

Table 2.3: Errors of Example 2.6.2 (an application of the linear finite difference method)

| $x_i$ | $w_i$ | $y(x_i)$ | $\|w_i - y(x_i)\|$ |
|---|---|---|---|
| 0. | -0.3 | -0.3 | |
| 0.16 | -0.311979 | -0.311950 | $2.9 \times 10^{-5}$ |
| 0.31 | -0.316260 | -0.316219 | $4.1 \times 10^{-5}$ |
| 0.47 | -0.312742 | -0.312701 | $4.1 \times 10^{-5}$ |
| 0.63 | -0.301513 | -0.301484 | $2.9 \times 10^{-5}$ |
| 0.79 | -0.282855 | -0.282843 | $1.2 \times 10^{-5}$ |
| 0.94 | -0.257231 | -0.257237 | $6.0 \times 10^{-6}$ |
| 1.10 | -0.225274 | -0.225298 | $2.4 \times 10^{-5}$ |
| 1.26 | -0.187778 | -0.187811 | $3.3 \times 10^{-5}$ |
| 1.41 | -0.145672 | -0.145699 | $2.7 \times 10^{-5}$ |
| 1.57 | -0.1 | -0.1 | |

The difference method for nonlinear problems given by (2.32) is similar to the method for linear problems discussed in the previous section. However, the method for nonlinear problems results in a nonlinear system of equations requiring an iterative method to be used.

Similar to the linear case, we first divide the domain [a, b] into $N+1$ subintervals whose endpoints are $x_i = a + ih$, for $i = 0, 1, ..., N+1$ where $h = (b-a)/(N+1)$. Therefore, the ODE given by (2.32) at the interior points, $x_i$, is given by

$$y''(x_i) = f(x_i, y(x_i), y'(x_i)),$$

Then we replace $y'(x_i)$ and $y''(x_i)$ by centered-difference formulas (2.26) and

(2.27) resulting in

$$\frac{y(x_{i+1})-2y(x_i)+y(x_{i-1})}{h^2}=f\left(x_i,y(x_i),\frac{y(x_{i+1})-y(x_{i-1})}{2h}-\frac{h^2}{6}y'''(\eta_i)\right)+\frac{h^2}{12}y^{(4)(\xi_i)}, \tag{2.33}$$

for some $\xi_i \in (x_{i-1}, x_{i+1})$.

As in the linear case, employing the boundary conditions and neglecting the error terms results in

$$w_0=\gamma_1, \quad w_{N+1}=\gamma_2,$$

and

$$-\frac{w_{i+1}-2w_i+w_{i-1}}{h^2}+f\left(x_i,w_i,\frac{w_{i+1}-w_{i-1}}{2h}\right)=0.$$

for i= 1, 2,..., N.

Finally to find approximate solutions, $w_i$, we need to solve the $(N \times N)$ nonlinear system given by

$$\begin{aligned}
0&=2w_1-w_2+h^2f\left(x_1,w_1,\frac{w_2-\gamma_1}{2h}\right)-\gamma_1,\\
0&=-w_1+2w_2-w_3+h^2f\left(x_2,w_2,\frac{w_3-w_1}{2h}\right),\\
&\vdots\\
0&=-w_{N-2}+2w_{N-1}-w_N+h^2f\left(x_{N-1},w_{N-1},\frac{w_N-w_{N-2}}{2h}\right),\\
0&=-w_{N-1}+2w_N+h^2f\left(x_N,w_N,\frac{\gamma_2-w_{N-1}}{2h}\right)-\gamma_2,
\end{aligned} \tag{2.34}$$

It is evidenced in reference [34] that the system given by (2.34) has a unique solution if $h < 2/L$. The solution to this nonlinear system is approximated by the Newton's method in section 11.4 of reference [9]. We skip repeating these details here. Therefore, we refer interested readers to the latter reference.

# Chapter 3

# Sinc Numerical Methods

## 3.1 Background

Sinc methods reduce the governing differential or integral equations to a system of algebraic equations, which makes the problem easier to solve. It is well-known that Sinc methods, developed by Frank Stenger and his colleagues [67], are dominant over other numerical methods, especially in the presence of singularities and semi-infinite domains [66]. They are also characterized by exponentially decaying errors and rapid convergence [32].

In general, there are two equivalent but distinct Sinc approaches: Sinc-Galerkin and Sinc-Collocation. A comprehensive description of these two approaches can be found in sections 3.7 and 3.8.

Sinc-based methods have been applied to diverse scientific and engineering problems comprising heat conduction [43, 49], population growth [4], inverse problems [45, 65], astrophysics problems [18, 54], medical imaging [68], elastoplastic problems [1], and oceanography models [36, 71]. Very recently, the application of the Sinc-Collocation approach to the telegraph equation [28] and the second type of the Painlevé equations [63] has been studied.

Sinc-methods have also been applied to systems of second-, third-, and fourth-order BVPs [16, 19, 61], fifth- and sixth-order BVPs arising in astrophysics [54], TPBVPs [8], Hallens integral equation [62], and integro-differential equations [47, 59].

The main goal of the current chapter is to describe Sinc methods fundamentals, approaches and their important properties in detail. We refer the interested readers to four references [37, 44, 66, 67], which cover the sinc methods in great detail.

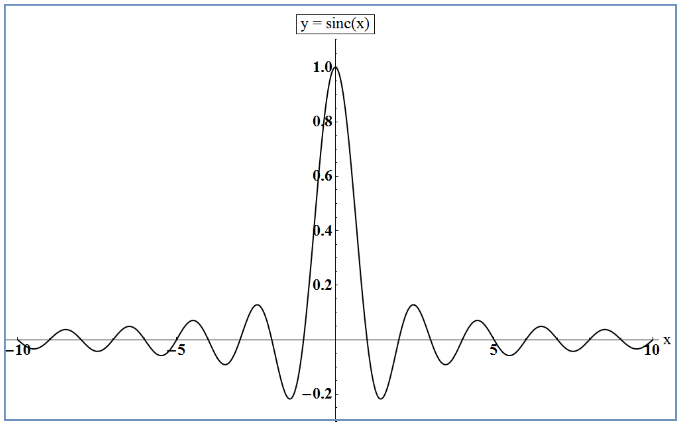
## 3.2 The Sinc Function Fundamentals

The function defined by

$$\operatorname{sinc}(z) \equiv \begin{cases} \frac{\sin(\pi z)}{\pi z} & \text{if } z \neq 0 \\ 1 & \text{if } z = 0 \end{cases} \tag{3.1}$$

for all $z \in C$, where $C = \{x + iy;\, x, y \in R,\, i^2 = -1\}$ is called the Sinc function. Even though the original Sinc function is defined on complex space, most of the time the translated Sinc function defined on the real line is

Figure 3.1: Sinc function



considered. The translated Sinc function is defined by;

$$\mathrm{S}(k,h)(x) \equiv \mathrm{sinc}\left(\frac{x-kh}{h}\right) \tag{3.2}$$

where $h > 0$, $k \in Z$ and $x \in R$.

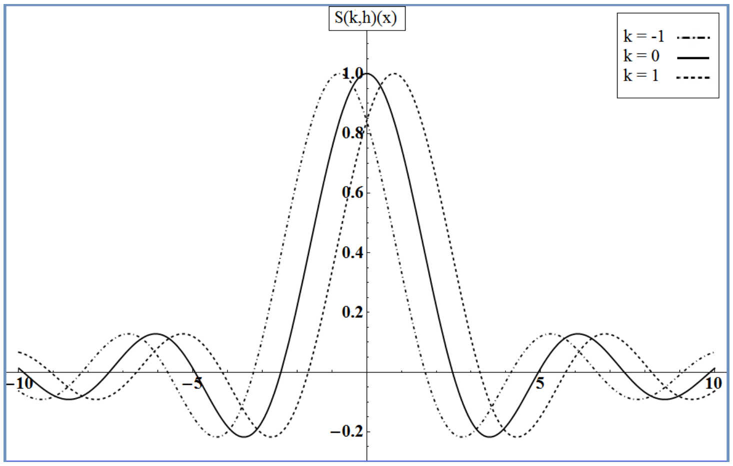
From (3.2), it follows that $\mathrm{sinc}(x) = \mathrm{S}(0,1)(x)$. Figure 3.1 portrays the plot of Sinc(x). As shown in Figure 3.1, $\mathrm{sinc}(0) = 1$ and for all other integers $\mathrm{sinc}(x)$ is equal to zero.

Since

$$\mathrm{S}(k,h)(x) \equiv \mathrm{sinc}\left(\frac{x-kh}{h}\right) \equiv \begin{cases} 1 & \text{if } x = kh \\ \frac{\sin(\pi(x-kh)/h)}{\pi(x-kh)/h} & \text{if } x \neq kh \end{cases} \tag{3.3}$$

for $k = 0, \pm 1, \pm 2, \pm 3, ...$, we can conclude that the translated Sinc function is similar to the standard Sinc function but shifted by a constant step size $h$ [22].

Figure 3.2: The translated Sinc function $S(k, \pi/4)(x)$ for k= -1, 0, 1.



For different values of $k$, the translated Sinc function $S(k, \pi/4)(x)$ on the whole real line is depicted in Figure 3.2. As well, for different values of step size, $h$, the central function $S(0, h)(x)$ is shown in Figure 3.3.

## 3.3 The Sinc Function Properties

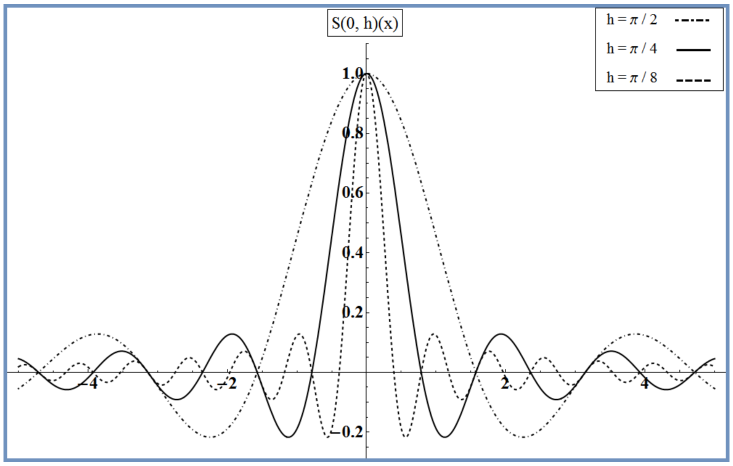
This section is devoted to some of the most important properties of the Sinc function. The interested reader may refer to the reference [44] for the proofs.

1. $\text{sinc}(z)$ is an entire function.

2. If $f(z) = \text{sinc}(z)$ then f $\in$ L$^2$(R), but f $\notin$ L$^1$(R).

3. For $h > 0$, consider the $\frac{2\pi}{h}$-periodic function given by $f(t) = e^{-izt}$,

Figure 3.3: The central function $S(0,h)(x)$ for h $= \pi/2, \pi/4, \pi/8$.



when $\frac{-\pi}{h} \leq t \leq \frac{\pi}{h}$. Then the Fourier series of $f(t)$ is given by

$$f(t) = \sum_{k=-\infty}^{\infty} e^{ikht} \operatorname{sinc}\left(\frac{z+hk}{h}\right). \tag{3.4}$$

4. The set $\{\frac{1}{\sqrt{h}} \operatorname{sinc}(\frac{t-kh}{h})\}_{k=-\infty}^{\infty}$ is orthogonal in $L^2$, that is

$$\int\limits_{-\infty}^{\infty} \operatorname{sinc}\left(\frac{t-kh}{h}\right) \operatorname{sinc}\left(\frac{t-jh}{h}\right) dt = h\, \delta_{kj}^{(0)}. \tag{3.5}$$

where

$$\delta_{kj}^{0} = \begin{cases} 0 & \text{if } k \neq j \\ 1 & \text{if } k = j \end{cases}. \tag{3.6}$$

5.

$$\int\limits_{-\infty}^{\infty} \operatorname{sinc}\left(\frac{z-kh}{h}\right) dz = h. \tag{3.7}$$

6. For $x \in R$, $j \in Z$, and $h > 0$ we have the identity

$$S(j,h)(x) = \frac{h}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{-ixt} e^{ijht} dt. \tag{3.8}$$

## 3.4 Exact Interpolation and Quadrature

The main idea of this section stems from the fact that for the appropriate class of functions $f$, a function which interpolates $f$ at a finite countable number of points on the real line, can be exact for all $z \in C$. Now we start this section with a fundamental definition.

**Definition 3.4.1**

Assuming that function $f$ is defined on the whole real line R and constant $h > 0$, the series (3.9) is called the Whittaker cardinal expansion of $f$, when it converges.

$$C(f,h)(x) \equiv \sum_{k=-\infty}^{\infty} f(kh) \operatorname{sinc}\left(\frac{x - kh}{h}\right). \tag{3.9}$$

The properties of this series is extensively discussed in [70].

Now we introduce a class of functions where the cardinal function of $f$ converges to $f$.

**Definition 3.4.2**

The Paley-Wiener class of functions $B(h)$, is the family of entire functions $f$ such that on the real line $f \in L^2(R)$ and in the complex plane $f$ is of exponential type $\frac{\pi}{h}$, i.e.,

$$|f(z)|\leq Kexp\left(\frac{\pi|z|}{h}\right) \tag{3.10}$$

for $h > 0$, $z \in C$ and some $K > 0$.

**Theorem 3.4.1** *(The Paley-Wiener Theorem)*

Consider an entire function $f \in L^2(R)$. If for all $z \in C$, there are positive constants $k$ and $\pi/h$ such that

$$|f(z)|\leq Kexp\left(\frac{\pi|z|}{h}\right) \tag{3.11}$$

then $\mathcal{F}(f) \in L^2(-\pi/h, \pi/h)$ and

$$f(z) = \frac{1}{2\pi}\int_{-\pi/h}^{\pi/h} \mathcal{F}(f)(x)exp(-ixz)dx. \tag{3.12}$$

Note that the notation $\mathcal{F}(f)(x)$ in (3.12) is the Fourier transform of $f$. If $f \in L^1(R)$, then the Fourier transform of $f$ is everywhere defined by the integral

$$\mathcal{F}(f) = \int_{-\infty}^{\infty} f(t)\exp(ixt)dt. \tag{3.13}$$

If $f \in L^2(R)$, then in general the integral has to be replaced by a limit and the Fourier transform is determined almost everywhere (a.e.).

**Theorem 3.4.2**

If $f \in B(h)$, then for all $z \in C$,

$$f(z) = \frac{1}{h} \int_{-\infty}^{\infty} f(t) \operatorname{sinc}\left(\frac{t-z}{h}\right) dt. \tag{3.14}$$

**Theorem 3.4.3**

If $g \in L^2(R)$, then $k \in B(h)$, where

$$k(z) \equiv \frac{1}{h} \int_{-\infty}^{\infty} g(t) \operatorname{sinc}\left(\frac{t-z}{h}\right) dt. \tag{3.15}$$

**Theorem 3.4.4**

If $f \in B(h)$, then for all $z \in C$,

$$f(z) = \sum_{-\infty}^{\infty} f(kh) \operatorname{sinc}\left(\frac{z-kh}{h}\right), \tag{3.16}$$

and

$$f(kh) = \frac{1}{h} \int_{-\infty}^{\infty} f(t) \operatorname{sinc}\left(\frac{t-kh}{h}\right) dt. \tag{3.17}$$

**Theorem 3.4.5** *(Interpolation Rule)*

If $f \in B(h)$, then for all $z \in C$

$$f(z) = C(f,h)(z) = \sum_{k=-\infty}^{\infty} f(kh) S(k,h)(z), \tag{3.18}$$

where

$$S(k,h)(z) \equiv \mathrm{sinc}\left(\frac{z-kh}{h}\right). \tag{3.19}$$

Proof: According to the third property of the sinc function, stated in section 3.3, the Fourier expansion of $f(t) = e^{-izt}$ is:

$$e^{-izt} = \sum_{k=-\infty}^{\infty} e^{ikht}\mathrm{sinc}\left(\frac{z+kh}{h}\right), \tag{3.20}$$

From the Paley-Wiener Theorem, we know that the inverse Fourier transform of f(z) is:

$$f(z) = \frac{1}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} F(f)(x)e^{-ixz}dx, \tag{3.21}$$

where

$$e^{-ixz} = \sum_{k=-\infty}^{\infty} e^{ikhx}\mathrm{sinc}\left(\frac{z+kh}{h}\right). \tag{3.22}$$

so

$$\begin{aligned} f(z) &= \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\mathrm{sinc}\left(\frac{z+kh}{h}\right)\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} F(f)(x)e^{-ikhx}dx \\ &= \sum_{k=-\infty}^{\infty}\mathrm{sinc}\left(\frac{z+kh}{h}\right)f(-kh) \\ &= \sum_{k=-\infty}^{\infty} f(kh)\mathrm{sinc}\left(\frac{z-kh}{h}\right). \end{aligned} \tag{3.23}$$

**Theorem 3.4.6** *(Quadrature Rule)*

If $f \in L^1(R)$, then

$$\int_{-\infty}^{\infty} f(z)dz = h\sum_{k=-\infty}^{\infty} f(kh). \tag{3.24}$$

Proof: By Integrating both sides of the interpolation formula given by (3.18), we have

$$\int_{-\infty}^{\infty} f(z)dz = \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(kh)\text{sinc}\left(\frac{z-kh}{h}\right) dz. \tag{3.25}$$

Since $f \in L^1(R)$,

$$\int_{-\infty}^{\infty} f(z)dz = \sum_{k=-\infty}^{\infty} f(kh) \int_{-\infty}^{\infty} \text{sinc}\left(\frac{z-kh}{h}\right) dz \tag{3.26}$$

Regarding the fifth property of the sinc function, stated in section 3.3,

$$\int_{-\infty}^{\infty} \text{sinc}\left(\frac{z-kh}{h}\right) dz = h,$$

Hence,

$$\int_{-\infty}^{\infty} f(z)dz = h \sum_{k=-\infty}^{\infty} f(kh). \tag{3.27}$$

## 3.5 Approximate Interpolation and Quadrature

In section 3.4, we introduced the Paley-Wiener class of functions for which the interpolation and quadrature formulas are exact and their approximation by cardinal series has errors which decrease exponentially. In this section, we define another less restrictive class of functions for which the interpolation and quadrature formulas are not exact but their approximation by cardinal functions still possess exponentially decaying errors. We begin this section with the definition of the desired class of functions and continue with some related definitions and theorems.

**Definition 3.5.1**

Consider the infinite strip domain of width $2d$, $d > 0$ denoted by $D_S$ (shown in Figure 3.4), and given by

$$D_S \equiv \{w \in C : w = u + iv, |v|< d\}. \tag{3.28}$$

Let $B^p(D_S)$ be the set of functions analytic in $D_S$ that satisfy

$$\int_{-d}^{d} |f(t+iv)|\, dv = O(|t|^a), \quad t \to \pm\infty, \;\; 0 \le a < 1 \tag{3.29}$$

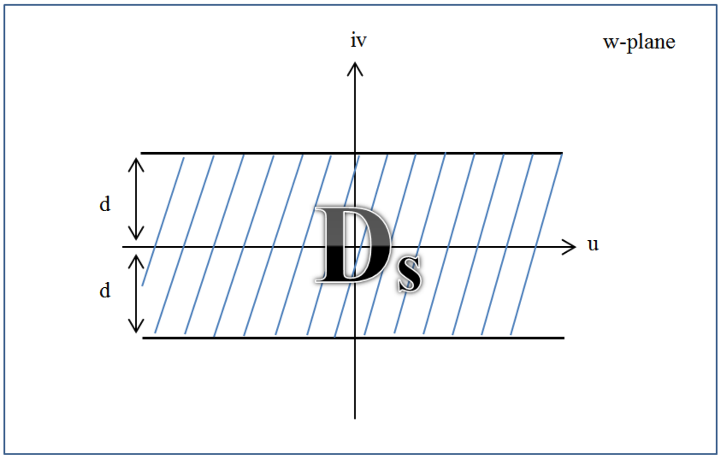
and

$$N^p(f, D_S) \equiv \lim_{v \to d^-} \left\{ \left( \int_{-\infty}^{\infty} |f(t+iv)|^p dt \right)^{1/p} + \left( \int_{-\infty}^{\infty} |f(t-iv)|^p dt \right)^{1/p} \right\} < \infty. \tag{3.30}$$

for $p = 1$, let $N(f, D_S) \equiv N^1(f, D_S)$ and $B(D_S) \equiv B^1(D_S)$.

The following theorem gives an interpolation result in $B^p(D_S)$.

Figure 3.4: The strip domain $D_S$.



**Theorem 3.5.1** *(Interpolation Rule)*

If $f \in B^p(D_S)$ (p = 1 or 2) and $h > 0$, then

$$f(x) - C(f,h)(x) = f(x) - \sum_{k=-\infty}^{\infty} f(kh)\,\text{sinc}\left(\frac{x-kh}{h}\right)$$

$$\equiv \epsilon(x) = S_h(x)\; I(f,h)(x).$$

where

$$S_h(x) \equiv \frac{\sin(\pi x/h)}{2\pi i}, \tag{3.31}$$

$$I(f,h)(x) \equiv \int_{-\infty}^{\infty} \left\{ \frac{F(x,t-id^-)}{\sin(\pi(t-id^-)/h)} - \frac{F(x,t+id^-)}{\sin(\pi(t+id^-)/h)} \right\} dt, \tag{3.32}$$

and

$$F(s, u \pm iv) = \frac{f(u \pm iv)}{(u - s \pm iv)}. \tag{3.33}$$

**Theorem 3.5.2** *(Quadrature Rule)*

If $f \in B^p(D_S)$, p = 1 or 2, then

$$\|f - C(f,h)\|_\infty \leq \frac{N^p(f, D_S)}{2(\pi d)^{1/p} \sinh(\pi d/h)} = \mathcal{O}(e^{-\pi d/h}). \tag{3.34}$$

The proofs of these theorems can be found in reference [44], pages 35-38.

## 3.6 Numerical Methods on an Arc $\Gamma$

Earlier we talked about numerical methods on the real line. Encountering problems with semi-infinite or finite domains, we require to map the domains to the infinite domain $D_S$, defined in (3.28). To fulfill this requirement we need to define a proper conformal map, denoted by $\varphi$. In this section, we introduce some of the most common conformal maps utilized in other studies.

For the semi-infinite domain (0, $\infty$), we may utilize;

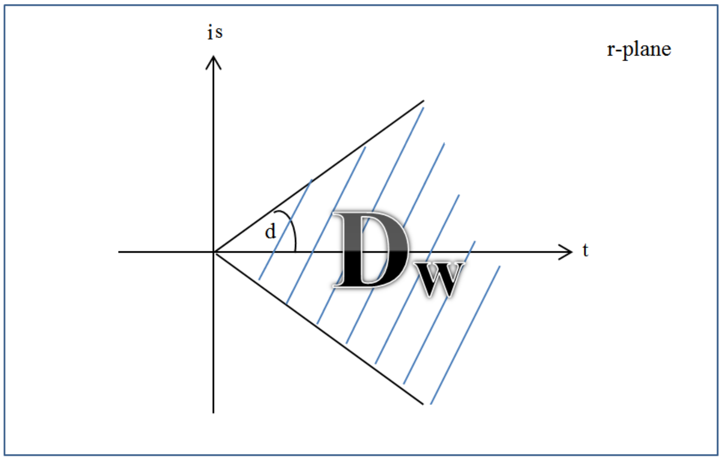
$$z = \varphi(w) = \ln(w),$$

or

$$z = \varphi(w) = \ln(\sinh(w)).$$

as well as the wedge-shaped domain, $D_W$, depicted in Figure 3.5, and given

Figure 3.5: The wedge domain $D_W$.



by

$$D_W \equiv \{w \in C : w = t + is, |argw|< d\}. \tag{3.35}$$

For the finite domain (0, 1), we may apply;

$$z = \varphi(w) = \ln\left(\frac{w}{1-w}\right)$$

and the eye-shaped domain, $D_E$, plotted in Figure 3.6, and given by

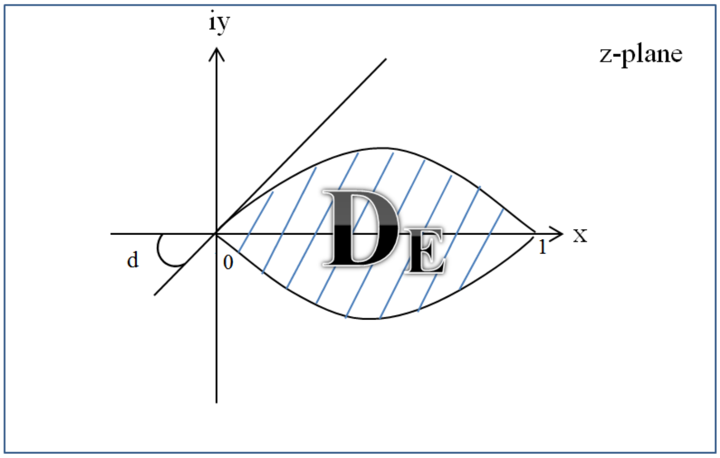
$$D_E = \{w \in C : w = x + iy, \left|arg\left(\frac{w}{1-w}\right)\right| < d\}. \tag{3.36}$$

Note that for any arbitrary finite interval (a, b) we can easily define the conformal map

$$z = \varphi(w) = \ln\left(\frac{w-a}{b-w}\right)$$

and the corresponding eye-shaped domain $D_E$.

Figure 3.6: The eye-shaped domain $D_E$.



The relationship between the eye-shaped domain, $D_E$, and the infinite strip, $D_S$, is shown in Figure 3.7. As well, we have shown the relationship between the wedge-shaped domain, $D_W$, and the infinite strip, $D_S$, in Figure 3.8.

## 3.7 The Sinc Collocation Method

Consider the boundary value problem

$$\mathcal{L}u(x) = -u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad a < x < b \tag{3.37}$$

with boundary conditions

$$u(a) = u(b) = 0.$$

Figure 3.7: The relationship between the eye-shaped domain $D_E$ and the infinite strip domain $D_S$.

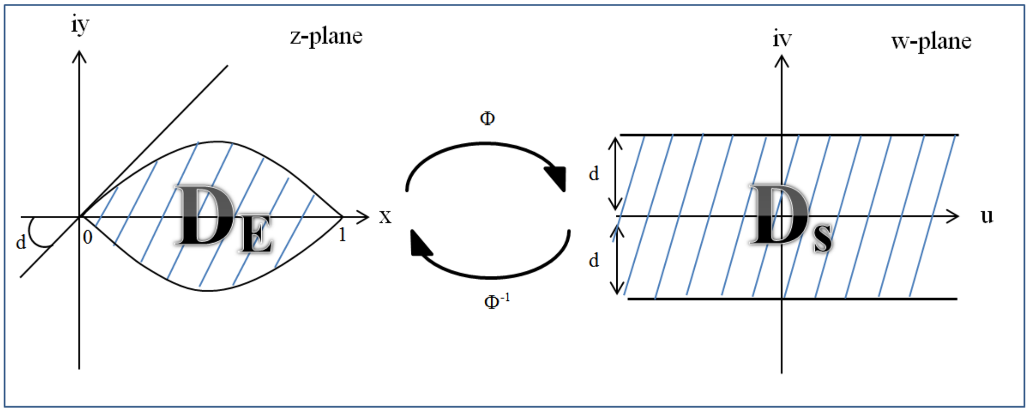
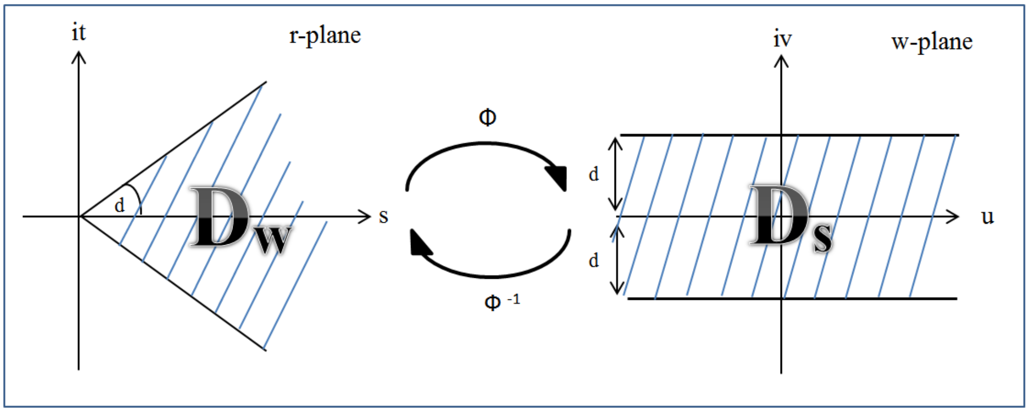


Figure 3.8: The relationship between the wedge-shaped domain $D_W$ and the infinite strip domain $D_S$.



A collocation scheme for (3.37) is defined by

$$\mathcal{L}u(x_j) = f(x_j), \quad a < x_j < b, \quad j = 1, 2, ..., m. \tag{3.38}$$

and has two main requirements. The first requirement is selecting the $m$ points $\{x_1 = a, ..., x_j, ..., x_m = b\}$, and the second is specifying the approximations to the derivatives $u'(x_j)$ and $u''(x_j)$ on the left-hand side of (3.38).

We assume that $u(x)$ is the exact solution of (3.37). According to the truncated interpolation rule, we can approximate the solution via

$$u(x) \simeq u_m(x) = \sum_{k=-M}^{N} u_k S(k,h) \circ \phi(x), \quad m = M + N + 1 \tag{3.39}$$

where

$$S(k,h) \circ \phi(x) = \operatorname{sinc}\left(\frac{\phi(x) - kh}{h}\right), \quad k = -M, ..., N. \tag{3.40}$$

Equation (3.39) provides an accurate approximation to the exact solution of equation (3.37). Even though the most direct approach to approximate $u'(x)$ and $u''(x)$ in equation (3.37) is by obtaining the derivatives of $u_m(x)$, it is not advised here. Indeed, differentiation of the sinc expansion of $u(x)$ on the entire real line possesses less restrictive assumptions than does differentiation of $u(x)$ on a subset of the real line [44]. Therefore, a variable transformation of $u(x)$ to the whole real line will be applied as

$$v(t) = (\phi'(\psi(t)))^l u(\psi(t)). \tag{3.41}$$

Where $\phi$ and $\psi$ are inverses of each other and $\phi$ is a one-to-one conformal map transferring the interval (a, b) into (-$\infty$, $\infty$) such that $\phi$(a)=-$\infty$, $\phi$(b)=$\infty$ and $l$ is a nonnegative real number [44]. Then using the chain rule and x=$\psi$(t) we have

$$\begin{aligned}
u(x) &= (\phi'(x))^{-l} v(x), \\
\frac{du}{dx} &= \frac{d}{dx}(\phi'(x)^{-l} v(x)) \\
&= (\phi'(x))^{-l}\frac{dv}{dx} + ((\phi'(x))^{-l})' v(x), \\
\frac{d^2u}{dx^2} &= (\phi'(x))^{-l}\frac{d^2v}{dx^2} + 2(\phi'(x)^{-l})'\frac{dv}{dx} + (\phi'(x)^{-l})'' v(x) \\
&= (\phi'(x))^{-l}\frac{}{d^2v}dx^2 - 2l(\phi'(x)^{-l-1})'\phi''(x)\frac{dv}{dx} + (\phi'(x)^{-l})'' v(x).
\end{aligned} \tag{3.42}$$

Substituting preceding equalities in (3.37) and multiplying it by $(\phi'(x))^l$ leads to

$$-\frac{d^2v}{dx^2} + \{\frac{2l\phi''(x)}{\phi'(x)} + p(x)\}\frac{dv}{dx}$$

$$+\{-(\phi'(x))^l((\phi'(x))^{-l})'' - \frac{l\phi''(x)p(x)}{\phi'(x)} + q(x)\}v(x)$$

$$= (\phi'(x))^l f(x).$$

Afterwards, the x-derivatives of the variable $v$ are replaced by

$$\begin{aligned} \frac{dv}{dx} &= \frac{dv}{dt}\frac{dt}{dx} = \phi'(x)\frac{dv}{dt}, \\ \frac{d^2v}{dx^2} &= \frac{d}{dx}(\phi'(x)\frac{dv}{dt}) = \phi''(x)\frac{dv}{dt} + (\phi'(x))^2\frac{d^2v}{dt^2}. \end{aligned} \tag{3.43}$$

Dividing the entire equation by $(\phi'(x))^2$ leads to

$$-\frac{d^2v}{dt^2} + \{(2l-1)\frac{\phi''(x)}{\phi'(x)^2} + \frac{p(x)}{\phi'(x)}\}\frac{dv}{dt} \tag{3.44}$$

$$+\{-(\phi'(x))^{l-2}((\phi'(x))^{-l})'' - l\frac{\phi''(x)p(x)}{(\phi'(x))^3} + \frac{q(x)}{(\phi'(x))^2}\}v$$

$$= (\phi'(x))^{l-2}f(x).$$

To generate the various coefficients of equation (3.44) in terms of variable $t$, one needs the repeated use of

$$\phi'(x) = (\psi'(t))^{-1},$$

and

$$\frac{dt}{dx} = \phi'(x) = \frac{1}{\psi'(t)},$$

leading to the transformed boundary value problem

$$-v''(t) + \xi_p(t)v'(t) + \gamma_q(t)v(t) = (\psi'(t))^{2-l}f(\psi(t)), \quad -\infty < t < \infty \tag{3.45}$$

$$\lim_{x\to\pm\infty} v(t) = 0.$$

where the coefficients of $v'(t)$ and $v(t)$ are given by

$$\xi_p(t) = \xi_p(\phi(x)) = (2l-1)\frac{\phi''(x)}{\phi'(x)^2} + \frac{p(x)}{\phi'(x)}, \tag{3.46}$$

and

$$\gamma_q(t) = \gamma(\phi(x)) = -(\phi'(x))^{l-2}((\phi'(x))^{-l})'' - l\frac{\phi''(x)p(x)}{(\phi'(x))^3} + \frac{q(x)}{(\phi'(x))^2}, \tag{3.47}$$

noting that $x = \psi(t)$.

Now that our transformed BVP is defined on the whole real line, the sinc approximation and its derivatives

$$v(t) \simeq v_m(t) = \sum_{k=-M}^{N} v_k S(k,h)(t), \qquad m = M+N+1, \tag{3.48}$$

and

$$\frac{d^n}{dt^n}v(t) = \sum_{k=-M}^{N} v_k \frac{d^n}{dt^n}S(k,h)(t), \qquad n = 1,2, \tag{3.49}$$

guarantee accurate approximations to $v(t)$, $v'(t)$, and $v''(t)$ as long as $v(t)$ satisfies the hypotheses of Theorem 3.7.1.

Substituting (3.48) and (3.49) in (3.45) and evaluating these approximations at $t = jh$, $-M \leq j \leq N$, yields the equation

$$-\sum_{k=-M}^{N} v(kh)\frac{d^2}{dt^2}S(k,h)(jh) + \xi_p(jh)\sum_{k=-M}^{N} v(kh)\frac{d}{dt}S(k,h)(jh) \tag{3.50}$$

$$+\gamma_q(jh)\sum_{k=-M}^{N} v(kh)S(k,h)(jh) = (\psi'(jh))^{2-l}f(\psi(jh)), \quad j=1,...,m.$$

Defining the derivatives of the translated sinc function evaluated at the sinc nodes by

$$\delta_{jk}^{(n)} \equiv h^n \frac{d^n}{dt^n}[S(j,h)(t)]\bigg|_{t=jh}, \quad n=0,1,2,...$$

which for n=0, 1, 2 yields

$$\delta_{jk}^{(0)} = \begin{cases} 1, & j=k \\ 0, & j\neq k \end{cases},$$

$$\delta_{jk}^{(1)} = \begin{cases} 0, & j=k \\ \frac{(-1)^{k-j}}{k-j}, & j\neq k \end{cases},$$

and

$$\delta_{jk}^{(2)} = \begin{cases} -\frac{\pi^2}{3}, & j=k \\ \frac{-2(-1)^{k-j}}{(k-j)^2}, & j\neq k \end{cases}.$$

equation (3.50) can be simplified as

$$-\sum_{k=-M}^{N}\frac{1}{h^2}\delta_{kj}^{(2)}v(kh) + \sum_{k=-M}^{N}\frac{\xi_p(jh)}{h}\delta_{kj}^{(1)}v(kh) + \sum_{k=-M}^{N}\gamma_q(jh)v(kh) \tag{3.51}$$

$$= (\psi'(jh))^{2-l}f(\psi(jh)), \quad j=1,...,m.$$

Therefore, the compact form of equation (3.51) is given by

$$\sum_{k=-M}^{N} \left\{ -\frac{1}{h^2}\delta_{jk}^{(2)} - \frac{1}{h}\xi_p(jh)\delta_{jk}^{(1)} + \gamma_q(jh) \right\} v(kh) = (\psi'(jh))^{2-l} f(\psi(jh)). \tag{3.52}$$

If we define the square matrix $I^{(n)}$ by

$$I^{(n)} = [\delta_{jk}^{(n)}], \quad n = 0, 1, 2$$

and diagonal matrix as

$$D(g) \equiv \begin{pmatrix} g(x_{-M}) & & & & \\ & \ddots & & & \\ & & g(x_0) & & \\ & & & \ddots & \\ & & & & g(x_N) \end{pmatrix}$$

then we can show that the algebraic form of the transformed BVP is give by

$$\{-\frac{1}{h^2}I^{(2)} - \frac{1}{h}D(\xi(t))I^{(1)} + D(\gamma(t))\}\overrightarrow{v} = D((\psi')^{2-l})\overrightarrow{f}. \tag{3.53}$$

Utilizing the transformation $\overrightarrow{v} = D((\phi'(x))^l)\overrightarrow{u}$, the value of the coefficients $u_k$ is obtainable by solving

$$\{-\frac{1}{h^2}I^{(2)} - \frac{1}{h}D(\xi(t))I^{(1)} + D(\gamma(t))\}D((\phi'(x))^l)\overrightarrow{u} = D((\phi'(x))^{l-2})\overrightarrow{f}. \tag{3.54}$$

In general, the development of the collocation scheme is valid for each choice of $l$. However, the choices $l = 0, \frac{1}{2}$, and 1 have particular capabilities.

**Theorem 3.7.1**

Consider $D$ as a region in the complex w-plane containing the boundary points $a$, and $b$ ($a \neq b$), and $z = \phi(w)$ as a one-to-one conformal map of the domain $D$ onto the infinite strip $D_S$, such that

$$\lim_{w \to a} \phi(w) = \infty, \quad \text{and} \quad \lim_{w \to b} \phi(w) = \infty.$$

Suppose $\frac{\phi' f}{g} \in B(D)$ and

$$\left| \frac{d^l}{ds^l} \left[ \frac{g(s) \sin(\frac{\pi\phi(s)}{h})}{2\pi i(\phi(w) - \phi(s))} \right] \right|_{w \in \partial D} \leq Kh^{-l}, \tag{3.55}$$

$$\left| \frac{d^l}{ds^l} \left[ g(s) \mathrm{sinc} \left( \frac{\phi(s) - kh}{h} \right) \right] \right| \leq Lh^{-l},$$

for all l = 0,1,...,n. Assume that

$$\left| \frac{f(s)}{g(s)} \right| \leq K \begin{cases} e^{\alpha s} & s \in \psi((-\infty, 0)) \\ e^{-\beta s} & s \in \psi([0, \infty)) \end{cases} ,$$

where $\alpha, \beta, K$ are positive constants and $w = \psi(z)$ is the inverse of the conformal mapping $z = \phi(x)$. If $N = ceil(\frac{\alpha}{\beta} M)$ and $0 < h = \sqrt{(\frac{\pi d}{\alpha M})} \leq \frac{2\pi d}{\ln 2}$, then

$$\frac{d^l f}{ds^l} = \sum_{k=-M}^{N} \frac{f(w_k)}{g(w_k)} \frac{d^l}{ds^l} \left[ g(s) \text{sinc} \left( \frac{\phi(s) - kh}{h} \right) \right] + \epsilon_M, \tag{3.56}$$

where $w_k = \phi(kh)$ and $|\epsilon_M| = O(M^{\frac{l+1}{2}} e^{-\sqrt{\pi d \alpha M}})$.

## 3.8 The Sinc Galerkin Method

Consider the boundary value problem

$$\mathcal{L}u(x) \equiv -u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \qquad a < x < b \tag{3.57}$$

with the boundary conditions

$$u(a) = u(b) = 0.$$

Similar to the Sinc-Collocation approach, the Sinc-Galerkin scheme approximates the exact solution of u(x) by the truncated sinc approximation given by

$$u(x) \simeq u_m(x) = \sum_{k=-M}^{N} u_k S(k,h) \circ \phi(x), \quad m = M + N + 1, \tag{3.58}$$

where

$$S(k,h)\circ\phi(x)=\operatorname{sinc}\left(\frac{\phi(x)-kh}{h}\right),\qquad k=-M,...,N. \tag{3.59}$$

The function $\phi(x)$ in (3.59), transforms the interval $(a,b)$ to the whole real line $R=(-\infty,\infty)$ and can be given by

$$\phi(x)=\ln\left(\frac{x-a}{b-x}\right).$$

Consequently, we can define

$$x_k=\phi^{-1}(kh)=\frac{be^{kh}+a}{e^{kh}+1}.$$

In the Galerkin approach the residual $(\mathcal{L}u_m-f)$ must be orthogonal to $S(j,h)\circ\phi$ in the weighted inner product defined by

$$(F,G)=\int\limits_a^b F(x)G(x)w(x)dx. \tag{3.60}$$

Choosing a proper weight function, $w(x)$, is dependent on the boundary conditions, the domain and the differential equation itself.

In the other words, the Sinc-Galerkin approach evaluates the unknown coefficients $\{u_k\}$ by orthogonalizing the residual $(\mathcal{L}u_m-f)$ to $S(j,h)\circ\phi$.

Therefore, we must have

$$(\mathcal{L}u_m - f, S(j,h)\circ\phi) = 0, \quad j = -M, ..., N \tag{3.61}$$

Regarding (3.57) and (3.60), we can rewrite (3.61) as

$$-\int_a^b u''(x)[S(j,h)\circ\phi]w(x)dx + \int_a^b p(x)u'(x)[S(j,h)\circ\phi]w(x)dx \tag{3.62}$$

$$+\int_a^b q(x)u(x)[S(j,h)\circ\phi]w(x)dx = \int_a^b f(x)w(x)dx.$$

Perhaps the most direct approach to solve equation (3.62) is by utilizing the quadrature rule. However, the existence of $u'(x)$ and $u''(x)$ may not yield quadrature formula that have the desired exponential convergence. To prevent this the Sinc-Galerkin approach uses integration by parts to eliminate the derivatives of $u(x)$ from the integrations in equation (3.62). Afterwards, the quadrature rule can be applied.

As discussed, each integral in (3.62) will be treated separately as follows.

$$\begin{aligned}(pu', S(j,h)\circ\phi) &= \int_a^b p(x)u'(x)[S(j,h)\circ\phi(x)]w(x)dx \\ &= B_{T_1} - \int_a^b u(x)(p[S(j,h)\circ\phi]w)'(x)dx \\ &= B_{T_1} - \int_a^b u(x)\bigg[[S(j,h)\circ\phi]'pw(x) + [S(j,h)\circ\phi](pw)'(x)\bigg]dx,\end{aligned} \tag{3.63}$$

where

$$B_{T_1} = (up[S(j,h)\circ\phi]w)(x)\Big|_a^b. \tag{3.64}$$

Similarly,

$$\begin{aligned}(u'', S(j,h)\circ\phi) &= \int_a^b u''(x)[S(j,h)\circ\phi(x)]w(x)dx \\ &= B_{T_2} + \int_a^b u(x)([S(j,h)\circ\phi]w)''(x) \\ &= B_{T_2} + \int_a^b \Big\{u(x)[S(j,h)\circ\phi(x)]''w(x) + 2u(x)[S(j,h)\circ\phi]'w'(x) \\ &\qquad +u(x)[S(j,h)\circ\phi(x)]w''(x)\Big\}dx,\end{aligned} \tag{3.65}$$

where

$$B_{T_2} = (u'[S(j,h)\circ\phi]w)(x)\Big|_a^b - (u([S(j,h)\circ\phi]w)')(x)\Big|_a^b. \tag{3.66}$$

And finally

$$(qu, S(j,h)\circ\phi) = \int_a^b q(x)u(x)[S(j,h)\circ\phi]w(x)dx. \tag{3.67}$$

Now we exploit the quadrature rule in Theorem 3.8.2, to approximate the integrals in (3.63), (3.65) and (3.67). To make the presentation of theses approximations more convenient and readable we define the notation

$$\delta_{jk}^{(n)} \equiv h^n \frac{d^n}{d\phi^n}[S(j,h)\circ\phi(x)]\Big|_{x=x_k}, \qquad n = 0,1,2,... \tag{3.68}$$

where $x_k = \phi^{-1}(kh)$. The following theorem, explicitly evaluate these quantities for $n = 0, 1, 2$.

**Theorem 3.8.1**

If $\phi$ is a conformal one-to-one map of the domain $D$ onto $D_S$, then

$$\delta_{jk}^{(0)} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases},$$

$$\delta_{jk}^{(1)} = \begin{cases} 0, & j = k \\ \frac{(-1)^{k-j}}{k-j}, & j \neq k \end{cases},$$

and

$$\delta_{jk}^{(2)} = \begin{cases} -\frac{\pi^2}{3}, & j = k \\ \frac{-2(-1)^{k-j}}{(k-j)^2}, & j \neq k \end{cases}.$$

Therefore, the inner products given by (3.63), (3.65) and (3.67) can be rewritten as

$$(pu', S(j,h) \circ \phi) = B_{T_1} - h \sum_{k=-M}^{N} u(x_k) w(x_k) \frac{\delta_{jk}^{(1)}}{h} - h \frac{u(x_j)(pw)'(x_j)}{\phi'(x_j)}, \quad (3.69)$$

Similarly,

$$(u'', S(j,h) \circ \phi) = B_{T_2} + h \sum_{k=-M}^{N} u(x_k) \Big[ \frac{1}{h^2} \delta_{jk}^{(2)} \phi'(x_k) w(x_k) + \quad (3.70)$$

$$\frac{\delta_{jk}^{(1)}}{h}\left(\frac{\phi''(x_k)w(x_k)}{\phi'(x_k)}+2w'(x_k)\right)+\delta_{jk}^{(0)}\frac{w''(x_k)}{\phi'(x_k)}\Bigg],$$

and finally,

$$(qu, S(j,h)\circ\phi) = h\frac{u(x_j)q(x_j)}{\phi'(x_j)}. \tag{3.71}$$

In addition we have,

$$(f, S(j,h)\circ\phi) = h\frac{f(x_j)}{\phi'(x_j)}. \tag{3.72}$$

Substituting (3.69)-(3.72) in (3.57) leads to

$$B_{T2}+\sum_{k=-M}^{N}\left[\frac{-1}{h^2}\delta_{jk}^{(2)}\phi'(x_k)w(x_k)-\frac{1}{h}\delta_{jk}^{(1)}(\frac{\phi''(x_k)w(x_k)}{\phi'(x_k)}+2w'(x_k))\right] \tag{3.73}$$

$$-\frac{w''(x_j)}{\phi'(x_j)}u_j+B_{T1}-\sum_{k=-M}^{N}\frac{1}{h}\delta_{jk}^{(1)}p(x_k)w(x_k)u_k-\frac{(pw)'(x_j)}{\phi'(x_j)}u_j$$

$$+\frac{q(x_j)w(x_j)}{\phi'(x_j)}u_j=\frac{f(x_j)w(x_j)}{\phi'(x_j)}.$$

Here we difine the weight function $w(x)$ by

$$w(x)=\frac{1}{\phi'(x)}$$

which makes $B_{T1}$ and $B_{T2}$ vanish.

To present an algebraic format of the system in (3.73), we define the vectors

$$\vec{u}\equiv(u_{-M},...,u_0,...,u_N)^T,$$

and

$$\vec{f} \equiv (f(x_M), ..., f(x_0), ..., f(x_N))^T,$$

the $m \times m$ $(m = M + N + 1)$ matrices

$$I^{(n)} \equiv [\delta_{jk}^{(n)}], \qquad n = 0, 1, 2$$

and the diagonal matrix

$$D(g) \equiv \begin{pmatrix} g(x_{-M}) & & & & \\ & \ddots & & & \\ & & g(x_0) & & \\ & & & \ddots & \\ & & & & g(x_N) \end{pmatrix}.$$

The matrix $I^{(0)}$ is the identity matrix. The matrices $I^{(1)}$ and $I^{(2)}$ are given by

$$I^{(1)} = \begin{pmatrix} 0 & -1 & \frac{1}{2} & \cdots & \frac{(-1)^{m-1}}{m-1} \\ 1 & & & & \vdots \\ -\frac{1}{2} & & \ddots & & \frac{1}{2} \\ \vdots & & & & -1 \\ \frac{(-1)^m}{m-1} & \cdots & -\frac{1}{2} & 1 & 0 \end{pmatrix},$$

and

$$I^{(2)} = \begin{pmatrix} -\frac{\pi^2}{3} & 2 & -\frac{2}{2^2} & \cdots & -\frac{2(-1)^{m-1}}{(m-1)^2} \\ 2 & & & & \vdots \\ -\frac{2}{2^2} & & \ddots & & -\frac{2}{2^2} \\ \vdots & & & & 2 \\ -\frac{2(-1)^{m-1}}{(m-1)^2} & \cdots & -\frac{2}{2^2} & 2 & -\frac{\pi^2}{3} \end{pmatrix}.$$

Therefore, the system in (3.73) takes the form

$$\begin{aligned} \bigg\{ -\frac{1}{h^2} I^{(2)} D(\phi' w) - \frac{1}{h} I^{(1)} D\bigg(\frac{\phi'' w}{\phi'} + 2w' + pw\bigg) - D\bigg(\frac{w'' + (pw)' - qw}{\phi'}\bigg)\bigg\} \vec{u} \\ = D\bigg(\frac{w}{\phi'}\bigg)\vec{f}. \end{aligned} \tag{3.74}$$

By solving the latter system, we obtain approximations to the exact solution $u(x)$.

**Theorem 3.8.2**

Consider $D$ as a region in the complex w-plane containing the boundary points $a$, and $b$ $(a \neq b)$, and $z = \phi(w)$ as a one-to-one conformal map of the domain $D$ onto the infinite strip $D_S$, such that

$$\lim_{w \to a} \phi(w) = \infty, \quad \text{and} \quad \lim_{w \to b} \phi(w) = \infty.$$

Suppose $f \in B(D)$, and

$$|f(s)| \leq K \begin{cases} e^{-\alpha|\phi(s)|}, & s \in \psi((-\infty, 0)) \\ e^{-\beta|\phi(s)|}, & s \in \psi([0, \infty)) \end{cases} , \tag{3.75}$$

where $\alpha, \beta$, and $K$ are positive constants and $w = \psi(z)$ is the inverse of the conformal mapping $z = \phi(w)$. If $N = \text{ceil}(\frac{\alpha}{\beta}M)$ and $0 < h = \sqrt{\frac{\pi d}{\alpha M}} \leq \frac{2\pi d}{\ln 2}$, then for all $s \in \psi(R)$

$$\int_{\psi(R)} f(s)ds = h \sum_{k=-M}^{N} \frac{f(w_k)}{\phi'(w_k)} + \eta_M. \tag{3.76}$$

where $w_k = \phi$ and $|\eta_M| = \mathcal{O}(e^{-\sqrt{2\pi d\alpha M}})$.

# Chapter 4

# Problem Formulation

In this section, the applications of predicted current patterns are shortly studied. A brief explanation of the model of interest is also added. We would refer interested readers to the reference [71], where the model originated. In addition, the readers who are interested in oceanography modelling are referred to the reference [31].

Almost 71% of the surface of the Earth is covered by water. Oceans play a significant role in regulating the weather and climate of the world. Coastal regions around the world are home to approximately three billion people, which is about half of the world's population. It is anticipated that by 2025 this population is likely to double [11].

As detailed on the website of the U.S. Department of National Ocean and Atmospheric Administration (NOAA)[1], predicted current patterns are

[1] http://www.noaa.gov

important to shipping, commercial fishing, recreational boating, and safety. Search-and-rescue personnel can also use this information to determine where the water may carry a missing person or floating object. This information is also critical to run hazardous material cleanup operations. Hazardous materials such as oil and fuel from tankers, typically remain on or near the waters surface, and travel with surface currents and winds. Finally, currents affect swimmers and fishers. Therefore, we can realize how important studying oceanographic patterns is.

Over the last century, physical oceanography has developed from a descriptive to an explanatory and predictive science. The oceanographic model presented here is found in the work of Winter et al. [71], and describes wind-driven currents in coastal regions and semi-enclosed seas when the eddy viscosity is a continuously differentiable fuction of depth.

For a better understanding of the model, some related physical concepts are defined in the next section.

## 4.1 Governing Physical Concepts

### 4.1.1 Tides

"Tides" are driven by the gravitational force of the moon and sun, and are characterized by water moving up and down over a long period of time.

### 4.1.2 Currents

The term “current” describes the motion of water. Oceanic currents are driven by several factors and are usually named by their source.

One of the sources of currents is the rise and fall of the tides. The currents created by tides occur near the shore and in bays and are called “tidal currents”. Tidal currents change in a very regular pattern and can be easily predicted in the future.

The second source of ocean currents is wind. Winds drive currents that are at or near the ocean’s surface, these currents are called “wind-driven currents”. Wind-driven currents are generally measured in centimeters per second (cm/s).

The ocean’s water density differences due to temperature (thermo) and salinity (haline) in different parts of the ocean drives currents at both deep and shallow ocean depths. These currents move much slower than tidal or surface currents. This process is called the “thermohaline” circulation.

### 4.1.3 Eddy Viscosity

The viscosity of a fluid is a measure of its resistance to gradual deformation by stresses. For liquids, it corresponds to the informal notion of “thickness”. For example, honey has a higher viscosity than does water. Eddy viscosity behaves like molecular viscosity but on a large scale.

### 4.1.4 Coriollis Effect

If the Earth did not rotate and remained stationary, the atmosphere would circulate between the poles (high pressure areas) and the equator (a low pressure area) in a simple circulatory pattern. But because the Earth rotates, circulating air is deflected. Instead of circulating in a closed loop, the air deflects toward the right in the Northern Hemisphere and toward the left in the Southern Hemisphere, resulting in curved paths. This deflection is called the Coriolis effect.

### 4.1.5 Ekman Spiral

The Ekman spiral occurs as a consequence of the Coriolis effect. Wind blowing on the surface of the ocean has the greatest effect on the surface. However, for the lower layers of the ocean to move they must be pushed by the friction between the layers of water above. The lower layer moves slower than the layer above. Like surface water, the deeper water is deflected by the Coriolis effect to the right in the Northern Hemisphere and to the left in the Southern Hemisphere. This results in a spiral shown in Figure 4.1.

## 4.2 Model Creation

In this section we provide a brief explanation of an oceanography model, found in the work of [71]. We would refer interested readers to the latter reference for detailed information.

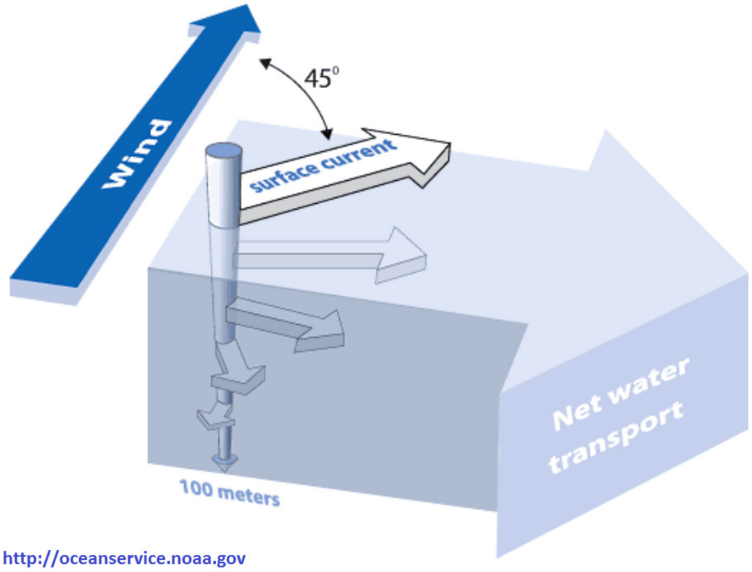


Figure 4.1: Ekman Spiral

To develop this model one needs to construct a right-handed coordinate system with the vertical coordinate $z^*$ directed positive downward from the free surface, and with $x^*$ and $y^*$ directed positive northward and eastward, respectively. It is supposed that $z^*$ changes from 0 to $D_0 = 100\,m$, and the plane at $z^* = D_0 = 100\,m$ is an impermeable boundary at the seabed [71]. This model is simplified by several assumptions. The ocean depth, $D_0$, and ocean mass density, $\rho$, are assumed constant, and the effects of tides, inertial terms, free surface slope, and variations in atmospheric pressure are neglected [71]. For a better understanding, a schematic form of the model is provided in Figure 4.2. The currents at the surface are driven by a tangential wind stress represented by $\tau(0) = \tau_w(\cos(\chi)\hat{x}^* + \sin(\chi)\hat{y}^*)$, where $\chi$ is the angle between the positive $x^*$-axis and the wind direction and $\hat{x}^*$ and $\hat{y}^*$ are unit

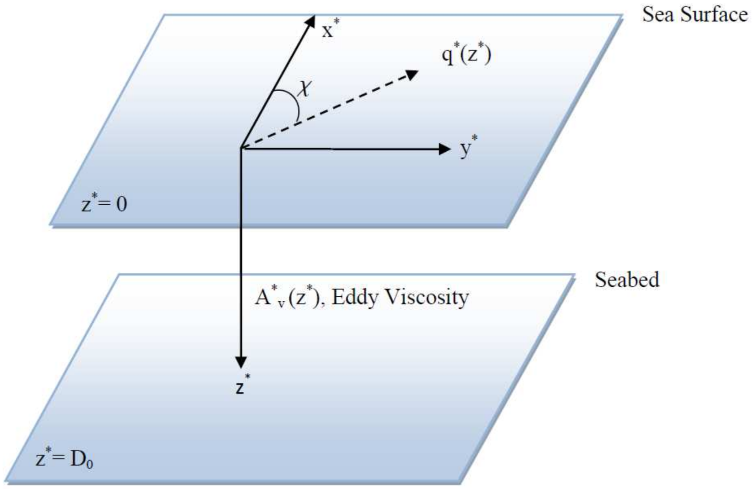


Figure 4.2: A schematic description of the oceanography model with depth-dependent eddy viscosity

vectors in the positive direction of $x^*$-axis and $y^*$-axis, respectively. The horizontal wind-drift current, $q^*(z^*)$, is given by $q^*(z^*) = U^*(z^*)\hat{x}^* + V^*(z^*)\hat{y}^*$. Internal frictional stresses are parameterized as $\tau(z^*) = -\rho A^*{}_v(z^*)\frac{dq}{dz^*}$, where the vertical eddy viscosity coefficient $A_v{}^*(z^*)$ is a continuously differentiable function of $z^* \in (0, D_0)$ [71].

With all these assumptions and considering the second Newton's law of motion the steady wind-drift current, $q^*$, will be driven by solving the boundary-value problem

$$\frac{d}{dz^*}(A^*{}_v(z^*)\frac{dq^*}{dz^*}) = -f\hat{z}^* \times q^*, \quad 0 < z^* < D_0, \tag{4.1}$$

where the stress condition at the sea surface, $z^* = 0$, is given by

$$-\rho {A_v}^*(0)\frac{dq^*(0)}{dz^*} = \tau_w \left(\cos(\chi)\hat{x}^* + \sin(\chi)\hat{y}^*\right), \tag{4.2}$$

and the condition at the seabed, $z^* = D_0$, is provided by

$$-\rho {A_v}^*(D_0)\frac{dq^*(D_0)}{dz^*} = k_f \; \rho \; q^*(D_0). \tag{4.3}$$

The Coriollis parameter at latitude $\theta$ is given by $f \equiv 2\Omega \sin(\theta)$, where $\Omega = 7.29 \times 10^{-5}$ rad s$^{-1}$ is the angular speed of rotation of the earth. Since the Coriollis force acts oppositely in the north and south hemisphere, it is assumed we are in northern hemisphere ($0 < \theta < \frac{\pi}{2}$). The parameter $k_f$ in (4.3) is the linear slip bottom stress coefficient.

By the definition of $q^*(z^*)$, the boundary-value problem in (4.1) could be simplified as

$$\begin{aligned}
\frac{d}{dz^*}\left({A^*}_v(z^*)\frac{dq^*}{dz^*}\right) &= \frac{d}{dz^*}\left({A^*}_v(z^*)\frac{dU^*(z^*)}{dz^*}\right)\hat{x}^* + \frac{d}{dz^*}\left({A^*}_v(z^*)\frac{dV^*(z^*)}{dz^*}\right)\hat{y}^* \\
&= -f\hat{z}^* \times q^* \\
&= -f\hat{z}^* \times [U^*(z^*)\hat{x}^* + V^*(z^*)\hat{y}^*] \\
&= -f\left(U^*(z^*)\hat{y}^* - V^*(z^*)\hat{x}^*\right), \quad 0 < z^* < D_0.
\end{aligned} \tag{4.4}$$

The equation (4.4) can be separated into components

$$-\frac{d}{dz^*}\left(A^*{}_v(z^*)\frac{dU^*(z^*)}{dz^*}\right) = -fV^*(z^*), \quad 0 < z^* < D_0, \tag{4.5}$$

and

$$-\frac{d}{dz^*}\left(A^*{}_v(z^*)\frac{dV^*(z^*)}{dz^*}\right) = -fU^*(z^*), \quad 0 < z^* < D_0. \tag{4.6}$$

As well, the boundary conditions at the sea surface, $z^* = 0$, and at the seabed, $z^* = D_0$, can be separated into

$$-\rho A^*{}_v(0)\frac{dU^*(0)}{dz^*} = \tau_w \cos(\chi), \qquad -\rho A^*{}_v(0)\frac{dV^*(0)}{dz^*} = \tau_w \sin(\chi), \tag{4.7}$$

and

$$-\rho A^*{}_v(D_0)\frac{dU^*(D_0)}{dz^*} = k_f \rho U^*(D_0), \qquad -\rho A^*{}_v(D_0)\frac{dV^*(D_0)}{dz^*} = k_f \rho V^*(D_0). \tag{4.8}$$

The equations of this model are nondimensionalized with the aid of a measure of near surface turbulent eddy viscosity, $A_0 \equiv A^*{}_v(0)$, and defining a nominal "upper-layer" Ekman depth by $D_E \equiv \sqrt{\frac{2A_0}{f}}$. In addition, the current speed in units of $U_0 = \frac{\tau_w D_E}{(\rho A_0)} = \frac{\sqrt{2}\tau_w}{(\rho\sqrt{A_0 f})}$ is defined. Here $U_0$ is the natural velocity scale in an infinitely deep sea with uniform eddy viscosity at steady state. With the help of the nondimensional variables

$$z \equiv \frac{z^*}{D_0}, \quad A_v(z) \equiv \frac{A^*{}_v(z^*)}{A^*{}_v(0)}, \quad q(z) \equiv \frac{q^*(z^*)}{U_0} \equiv U(z)\hat{x} + V(z)\hat{y}, \tag{4.9}$$

and nondimensional constants, $\kappa$ (depth ratio) and $\sigma$ (bottom friction parameter)

$$\kappa \equiv \frac{D_0}{D_E} = D_0\sqrt{\frac{f}{2A_0}}, \qquad \sigma \equiv \frac{A_0 A_v(1)}{k_f D_0} = \frac{A^*{}_v(D_0)}{k_f D_0}, \tag{4.10}$$

equations (4.5), and (4.6) are transformed to nondimensional equations, given by

$$-\frac{d}{dz}\left(A_v(z)\frac{dU(z)}{dz}\right) = -2\kappa^2 V(z), \qquad 0 < z < 1, \tag{4.11}$$

and

$$-\frac{d}{dz}\left(A_v(z)\frac{dV(z)}{dz}\right) = 2\kappa^2 U(z), \qquad 0 < z < 1. \tag{4.12}$$

In addition, the nondimensionalizing procedure applied to the boundary conditions leads to

$$\frac{dU(0)}{dz} = -\kappa\cos(\chi), \qquad \frac{dV(0)}{dz} = -\kappa\sin(\chi), \tag{4.13}$$

and

$$U(1) + \sigma\frac{dU(1)}{dz} = 0, \qquad V(1) + \sigma\frac{dV(1)}{dz} = 0. \tag{4.14}$$

The no-slip bottom condition can easily be driven by setting $\sigma = 0$ in equation (4.14).

To transform the nonhomogeneous boundary conditions to homogeneous ones, the following linear transformations are applied,

$$U(z) = u(z)+\kappa(1+\sigma-z)\cos(\chi), \quad V(z) = v(z)+\kappa(1+\sigma-z)\sin(\chi). \tag{4.15}$$

The first derivatives of the transformations in (4.15) are given by

$$\frac{dU(z)}{dz} = \frac{du(z)}{dz} - \kappa\cos(\chi), \quad \frac{dV(z)}{dz} = \frac{dv(z)}{dz} - \kappa\sin(\chi). \tag{4.16}$$

Hence the "reduced velocity" components $u(z)$ and $v(z)$ satisfy

$$-\frac{d}{dz}\left(A_v(z)\frac{du}{dz}\right)+\kappa\cos(\chi)A_v'(z) = -2\kappa^2 v(z)-2\kappa^3(1+\sigma-z)\sin(\chi), \quad 0 < z < 1, \tag{4.17}$$

and

$$-\frac{d}{dz}\left(A_v(z)\frac{dv}{dz}\right)+\kappa\sin(\chi)A_v'(z) = 2\kappa^2 u(z)+2\kappa^3(1+\sigma-z)\cos(\chi), \quad 0 < z < 1. \tag{4.18}$$

where the boundary conditions at the surface and seabed become

$$\frac{du(0)}{dz} = 0, \quad \frac{dv(0)}{dz} = 0, \tag{4.19}$$

and

$$u(1) + \sigma\frac{du(1)}{dz} = 0, \quad v(1) + \sigma\frac{dv(1)}{dz} = 0. \tag{4.20}$$

The above system can be shown in two formats; the coupled $u$ and $v$ equation system and complex velocity system. For constructing the coupled system, we define

$$\mathcal{L}u(z) \equiv -\frac{d}{dz}\left(A_v(z)\frac{du}{dz}\right), \quad \mathcal{L}v(z) \equiv -\frac{d}{dz}\left(A_v(z)\frac{dv}{dz}\right). \tag{4.21}$$

Therefore, (4.17) and (4.18) can be rewritten as

$$\mathcal{L}u(z) + 2\kappa^2 v(z) = F_1(z), \quad 0 < z < 1 \tag{4.22}$$

and

$$\mathcal{L}v(z) - 2\kappa^2 u(z) = F_2(z), \quad 0 < z < 1 \tag{4.23}$$

where

$$F_1(z) = -2\kappa^3(1+\sigma-z)\sin(\chi) - \kappa\cos(\chi)A_v'(z), \tag{4.24}$$

and

$$F_2(z) = 2\kappa^3(1+\sigma-z)\cos(\chi) - \kappa\sin(\chi)A_v'(z), \tag{4.25}$$

and boundary conditions at the surface and seabed are given respectively by

$$\frac{du(0)}{dz} = 0, \quad \frac{dv(0)}{dz} = 0, \tag{4.26}$$

and

$$u(1) + \sigma\frac{du(1)}{dz} = 0, \quad v(1) + \sigma\frac{dv(1)}{dz} = 0. \tag{4.27}$$

To construct the complex velocity formulation, one needs to multiply equation (4.18) by i, and add the result to equation (4.17). Then by defining a complex velocity $w(z) = u(z) + iv(z)$, we have

$$\begin{aligned}\mathcal{L}w(z) &\equiv \mathcal{L}u(z) + i\mathcal{L}v(z)\\ &\equiv -\frac{d}{dz}\left(A_v(z)\frac{du(z)}{dz}\right) - i\frac{d}{dz}\left(A_v(z)\frac{dv(z)}{dz}\right)\\ &\equiv -\frac{d}{dz}\left(A_v(z)\frac{dw(z)}{dz}\right).\end{aligned}$$

Hence, the coupled problem can be transformed to the complex velocity form

$$\mathcal{L}w(z) - i2\kappa^2 w(z) = F(z), \;\; 0 < z < 1, \tag{4.28}$$

where

$$\begin{aligned}F(z) &\equiv 2\kappa^3[i(1+\sigma-z)\cos(\chi) - (1+\sigma-z)\sin(\chi)] - \kappa A_v'(z)[\cos(\chi) + i\sin(\chi)]\\ &= [-\kappa A_v'(z) + i2\kappa^3(1+\sigma-z)]e^{i\chi}.\end{aligned}$$

Performing the same procedure to the boundary conditions leads to the surface condition

$$w'(0) = 0, \tag{4.29}$$

and the seabed condition

$$w(1) + \sigma w'(1) = 0. \tag{4.30}$$

# Chapter 5

# Our Sinc-Collocation Approach

In this section, we provide extensions of a new Sinc-Collocation approach found in the work of Abdella [2], and apply it to the oceanography model we discussed earlier in both complex and coupled systems.

## 5.1 Sinc-Collocation Solution of the Complex Velocity System

Assume the complex velocity problem given by:

$$\mathcal{L}w(z) - i2\kappa^2 w(z) = F(z), \;\; 0 < z < 1. \tag{5.1}$$

with boundry conditions

$$w'(0) = 0, \tag{5.2}$$

and

$$w(1) + \sigma w'(1) = 0. \tag{5.3}$$

where

$$\begin{aligned}\mathcal{L}w(z) &\equiv \mathcal{L}u(z) + i\mathcal{L}v(z)\\ &\equiv -\frac{d}{dz}\left(A_v(z)\frac{du(z)}{dz}\right) - i\frac{d}{dz}\left(A_v(z)\frac{dv(z)}{dz}\right)\\ &\equiv -\frac{d}{dz}\left(A_v(z)\frac{dw(z)}{dz}\right),\end{aligned}$$

and

$$\begin{aligned}F(z) &\equiv 2\kappa^3[i(1+\sigma-z)\cos(\chi) - (1+\sigma-z)\sin(\chi)] - \kappa A_v'(z)[\cos(\chi) + i\sin(\chi)]\\ &= [-\kappa A_v'(z) + i2\kappa^3(1+\sigma-z)]e^{i\chi}.\end{aligned}$$

To follow the desired Sinc-Collocation approach found in [2], we transform the boundary value problem as follows:

$$\eta(z) = w(z) - P(z), \tag{5.4}$$

where

$$P(z) = w'(a)H_1 + w(a)H_2 + w(b)H_3 + w'(b)H_4, \tag{5.5}$$

is the univariate Hermite interpolation with the cardinal functions given by:

$$H_1 = \frac{(z-a)(z-b)^2}{(b-a)^2},\ H_2 = \frac{(z-b)^2(2z-3a+b)}{(b-a)^3},$$

$$H_3 = \frac{(z-a)^2(2z-3b+a)}{(a-b)^3},\ H_4 = \frac{(z-b)(z-a)^2}{(b-a)^2}.$$

Employing (5.4) and considering $P(a) = w(a), P'(a) = w'(a), P(b) = w(b), P'(b) = w'(b)$, leads to a new BVP given by:

$$a(z)\eta''(z) + b(z)\eta'(z) + c(z)\eta(z) + \Lambda(z) = F(z),\ z \in (a,b), \tag{5.6}$$

$$\eta(a) = \eta(b) = 0,$$

$$\eta'(a) = \eta'(b) = 0,$$

where

$$\Lambda(z) = w'(a)\lambda_1(z) + w(a)\lambda_2(z) + w(b)\lambda_3(z) + w'(b)\lambda_4(z),$$

in which

$$\lambda_1(z) = a(z)H_1'' + b(z)H_1' + c(z)H_1,$$

$$\lambda_2(z) = a(z)H_2'' + b(z)H_2' + c(z)H_2,$$

$$\lambda_3(z) = a(z)H_3'' + b(z)H_3' + c(z)H_3,$$

$$\lambda_4(z) = a(z)H_4'' + b(z)H_4' + c(z)H_4,$$

and

$$a(z) = -A_v(z),\;\; b(z) = -A_v'(z), \text{ and } c(z) = -2\kappa^2 i.$$

Here is the point that our method changes from the original sinc-collocation technique. We first approximate $\eta'(x)$ at sinc points $z_i$ by:

$$\eta'(z_i) = \sum_{k=-N}^{N} S(k,h)(\varphi(z_i))\eta'(z_k) = \sum_{k=-N}^{N} \delta_{i,k}^{(0)}\eta'(z_k). \tag{5.7}$$

Afterwards, $\eta(z)$ is approximated by

$$\eta(z_i) = \sum_{k=-N}^{N} h_k(z_i)\eta'(z_k) = \sum_{k=-N}^{N} h\delta_{i,k}^{(-1)}\frac{\eta'(z_k)}{\varphi'(z_k)}. \tag{5.8}$$

Finally, we approximate $\eta''(z)$ via

$$\eta''(z_i) = \sum_{k=-N}^{N} g_k(z_i)\eta'(z_k) = \sum_{k=-N}^{N} \delta_{i,k}^{(1)}\varphi'(z_i)\frac{\eta'(z_k)}{h} \tag{5.9}$$

where $\varphi(z)$ is given by

$$\xi = \varphi(z) = \frac{1}{\pi}\log\left(\frac{z-a}{b-z}\right), \tag{5.10}$$

with inverse

$$z = \psi(\xi) = \frac{b+a}{2} + \frac{b-a}{2}\tanh\left(\frac{\pi}{2}\sinh(\xi)\right), \tag{5.11}$$

and $z_k = \psi(kh)$. Coefficients $\delta_{i,k}^{(n)}$ for n=0, 1, and 2 are defined by

$$\delta_{i,k}^{(0)} = \begin{cases} 0, & i \neq k \\ & \\ 1, & i = k, \end{cases} , \tag{5.12}$$

$$\delta_{i,k}^{(1)} = \begin{cases} \frac{(-1)^{i-k}}{i-k}, & k \neq i \\ & \\ 0, & k = i, \end{cases} , \tag{5.13}$$

and

$$\delta_{i,k}^{(-1)} = \begin{cases} \frac{1}{2} + \int_0^{i-k} \frac{\sin(\pi t)}{\pi t}, & i \neq k \\ & \\ \frac{1}{2}, & k = i. \end{cases} . \tag{5.14}$$

Hence, the discretized version of the equation (5.6) is given by

$$\sum_{k=-N}^{N} M_{i,k}\eta'(z_k) + \Lambda(z_i) = F(z_i), \quad i = -N, ..., N \tag{5.15}$$

where

$$M_{i,k} = a(z_i)\delta_{k,i}^{(1)} \frac{\varphi'(z_i)}{h} + b(z_i)\delta_{k,i}^{(0)} + c(z_i)h\frac{\delta_{k,i}^{(-1)}}{\varphi'(z_k)}. \tag{5.16}$$

Note that (5.15) leads to a system of $(n = 2N+1)$ equations for $(m = 2N+5)$ unknowns including $w'(a)$, $w(a)$, $w'(b)$, $w(b)$ and $\eta'(z_i), i = -N, ..., N$.

We define the $(2N+5)\times 1$ vector $\mathbf{C}$ by:

$$\mathbf{C} = [C_{-N-2}, C_{-N-1}, C_{-N}, ..., C_0, ..., C_N, C_{N+1}, C_{N+2}]^{\mathrm{T}}$$

$$= [w(a), w'(a), \eta'(z_{-N}), ..., \eta'(z_0), ..., \eta'(z_N), w'(b), w(b)]^{\mathrm{T}}.$$

The four conditions required to close the system are given by:

$$\alpha_a C_{-N-2} + \beta_a C_{-N-1} = \gamma_a, \tag{5.17}$$

$$\alpha_b C_{N+2} + \beta_b C_{N+1} = \gamma_b, \tag{5.18}$$

$$\sum_{k=-N}^{N} h\delta_{-N-1,k}^{(-1)} \frac{C_k}{\phi'(z_k)} = 0, \tag{5.19}$$

$$\sum_{k=-N}^{N} h\delta_{N+1,k}^{(-1)} \frac{C_k}{\phi'(z_k)} = 0. \tag{5.20}$$

The matrix representation of the $(2N+5)\times(2N+5)$ system corresponding to equations (5.15) and (5.17)-(5.20) is given by

$$\mathbf{AC} = \mathbf{F} \tag{5.21}$$

where $\mathbf{F}$, a $(2N+5)\times 1$ vector, and $\mathbf{A}$, a $(2N+5)\times(2N+5)$ matrix are given by

$$\mathbf{F} = [\gamma_a, \gamma_b, F(z_{-N}), ..., F(z_0), ..., F(z_N), 0, 0]^{\mathrm{T}},$$

$$\mathbf{A} = \begin{pmatrix} B_1 \\ B_2 \\ \mathbf{B} \\ B_3 \\ B_4 \end{pmatrix}, \tag{5.22}$$

where $B_1$, $B_2$, $B_3$ and $B_4$ are $1 \times (2N+5)$ matrices given by

$$B_1 = [\alpha_a, \beta_a, 0, ..., 0],$$

$$B_2 = [0, 0, ..., \beta_b, \alpha_b],$$

$$B_3 = [0, 0, \frac{h\delta^{(-1)}_{-N-1,-N}}{\phi'(z_k)}, ..., \frac{h\delta^{(-1)}_{-N-1,N}}{\phi'(z_k)}, 0, 0],$$

$$B_4 = [0, 0, \frac{h\delta^{(-1)}_{N+1,-N}}{\phi'(z_k)}, ..., \frac{h\delta^{(-1)}_{N+1,N}}{\phi'(z_k)}, 0, 0],$$

and $\mathbf{B}$ as a $(2N+1) \times (2N+5)$ matrix is given by

$$\mathbf{B} = [\lambda_2^T, \lambda_1^T, \mathbf{M}, \lambda_4^T, \lambda_3^T],$$

where $\mathbf{M}$ is the $(2N+1) \times (2N+1)$ matrix format of (5.16). Please note that $\lambda_i^T$, is the transpose of $\lambda_i$.

Once equation (5.21) is solved, the coefficients are used to determine the unknown function $\eta(z)$ and its first and second derivatives at the Sinc nodes using equations (5.7)-(5.9). The original unknown, $w(z)$ is then determined from equation (5.4). Note that the values of $w(z)$ and $w'(z)$ at the two endpoints are also determined directly from the system solutions. The unknowns $u(z)$ and $v(z)$ are the real and imaginary parts of $w(z)$ respectively, obtained via

$$u(z) = Re[w(z)],$$

and

$$v(z) = Im[w(z)].$$

u(z) and v(z) are reduced velocities of the current in north and east directions. To obtain U(z) and V(z), we need to apply the equations given in (4.15).

## 5.2 Sinc-Collocation Solution of the Coupled Differential Equation System

The model of interest is given by the following coupled system of differential equations

$$\mathcal{L}u(z) + 2\kappa^2 v(z) = F_1(z), \quad 0 < z < 1, \tag{5.23}$$

$$\mathcal{L}v(z) - 2\kappa^2 u(z) = F_2(z), \quad 0 < z < 1, \tag{5.24}$$

with boundary conditions

$$\frac{du(0)}{dz} = 0, \qquad \frac{dv(0)}{dz} = 0, \tag{5.25}$$

$$u(1) + \sigma\frac{du(1)}{dz} = 0, \qquad v(1) + \sigma\frac{dv(1)}{dz} = 0. \tag{5.26}$$

where

$$\mathcal{L}u(z) \equiv -\frac{d}{dz}\left(A_v(z)\frac{du}{dz}\right), \tag{5.27}$$

$$\mathcal{L}v(z) \equiv -\frac{d}{dz}\left(A_v(z)\frac{dv}{dz}\right), \tag{5.28}$$

$$F_1(z) = -2\kappa^3(1+\sigma-z)\sin(\chi) - \kappa\cos(\chi)A_v'(z), \tag{5.29}$$

$$F_2(z) = 2\kappa^3(1+\sigma-z)\cos(\chi) - \kappa\sin(\chi)A_v'(z). \tag{5.30}$$

To follow the desired Sinc-Collocation approach found in [2], we transform the boundary value problem as follows:

$$y_u(z) = u(z) - P_u(z), \tag{5.31}$$

$$y_v(z) = v(z) - P_v(z), \tag{5.32}$$

where

$$P_u(z) = u'(a)H_1 + u(a)H_2 + u(b)H_3 + u'(b)H_4, \tag{5.33}$$

$$P_v(z) = v'(a)H_1 + v(a)H_2 + v(b)H_3 + v'(b)H_4, \tag{5.34}$$

are the univariate Hermite interpolations with the cardinal functions given by:

$$H_1 = \frac{(z-a)(z-b)^2}{(b-a)^2},\ H_2 = \frac{(z-b)^2(2z-3a+b)}{(b-a)^3},$$

$$H_3 = \frac{(z-a)^2(2z-3b+a)}{(a-b)^3},\ H_4 = \frac{(z-b)(z-a)^2}{(b-a)^2}.$$

Regarding $P_u(a) = u(a), P'_u(a) = u'(a), P_u(b) = u(b), P'_u(b) = u'(b)$, and $P_v(a) = v(a), P'_v(a) = v'(a), P_v(b) = v(b), P'_v(b) = v'(b)$, the above BVP is transformed into the following problem

$$a(z)y''_u(z) + b(z)y'_u(z) + c_1(z)y_v(z) + \Lambda_1(z) = F_1(z), \tag{5.35}$$

$$a(z)y''_v(z) + b(z)y'_v(z) + c_2(z)y_u(z) + \Lambda_2(z) = F_2(z), \tag{5.36}$$

$$y_u(a) = y_u(b) = 0, \quad y_v(a) = y_v(b) = 0,$$

$$y'_u(a) = y'_u(b) = 0, \quad y'_v(a) = y'_v(b) = 0,$$

where

$$\Lambda_1(z) = \eta_1 u'(a) + \eta_2 u(a) + \eta_3 u(b) + \eta_4 u'(b) + \zeta_1 v'(a) + \zeta_2 v(a) + \zeta_3 v(b) + \zeta_4 v'(b),$$

$$\Lambda_2(z) = \eta_1 v'(a) + \eta_2 v(a) + \eta_3 v(b) + \eta_4 v'(b) + \zeta'_1 u'(a) + \zeta'_2 u(a) + \zeta'_3 u(b) + \zeta'_4 u'(b),$$

in which

$$\eta_1 = a(z)H''_1 + b(z)H'_1, \quad \eta_2 = a(z)H''_2 + b(z)H'_2,$$

$$\eta_3 = a(z)H_3'' + b(z)H_3', \quad \eta_4 = a(z)H_4'' + b(z)H_4',$$

$$\zeta_1 = c_1(z)H_1, \quad \zeta_2 = c_1(z)H_2,$$

$$\zeta_3 = c_1(z)H_3, \quad \zeta_4 = c_1(z)H_4,$$

$$\zeta_1' = c_2(z)H_1, \quad \zeta_2' = c_2(z)H_2,$$

$$\zeta_3' = c_2(z)H_3, \quad \zeta_4' = c_2(z)H_4.$$

and

$$a(z) = -A_v(z), \;\; b(z) = -A_v'(z), \;\; c_1(z) = 2\kappa^2, \;\; c_2(z) = -2\kappa^2,$$

In this Sinc-Collocation approach we first approximate $y_u'(z)$ and $y_v'(z)$ by:

$$y_u'(z) \approx \sum_{k=-N}^{N} S(k,h)(\varphi(z))y_u'(z_k), \;\; a \le z \le b, \tag{5.37}$$

and

$$y_v'(z) \approx \sum_{k=-N}^{N} S(k,h)(\varphi(z))y_v'(z_k), \;\; a \le z \le b, \tag{5.38}$$

where $\varphi(z)$ is the double exponential transformation

$$\xi = \varphi(z) = \log(G) + \sqrt{G^2+1}, \; G = \frac{1}{\pi}\log\left(\frac{z-a}{b-z}\right) \tag{5.39}$$

with inverse

$$z = \psi(\xi) = \frac{b+a}{2} + \frac{b-a}{2} \tanh\left(\frac{\pi}{2}\sinh(\xi)\right) \tag{5.40}$$

and $z_k = \psi(kh)$. Hence the unknown variables $y_u(z)$ and $y_v(z)$ can be approximated by:

$$y_u(z) = \int_a^z y'_u(z) + y_u(a) = \sum_{k=-N}^{N} h_k(z) y'_u(z_k), \;\; a \le z \le b, \tag{5.41}$$

$$y_v(z) = \int_a^z y'_v(z) + y_v(a) = \sum_{k=-N}^{N} h_k(z) y'_v(z_k), \;\; a \le z \le b,$$

where

$$h_k(z) = \int_a^z S(k,h)(\varphi(z))dz. \tag{5.42}$$

Similarly, we can approximate $y''_u(z)$ and $y''_v(z)$ by differentiation:

$$y''_u(z) = \sum_{k=-N}^{N} g_k(z) y'_u(z_k), \;\; a \le z \le b, \tag{5.43}$$

and

$$y''_v(z) = \sum_{k=-N}^{N} g_k(z) y'_v(z_k), \;\; a \le z \le b.$$

where

$$g_k(z) = \frac{dS(k,h)(\varphi(z))}{dz}. \tag{5.44}$$

Therefore, at the Sinc points $z_i$ we have:

$$y'_u(z_i) = \sum_{k=-N}^{N} S(k,h)(\varphi(z_i))y'_u(z_k) = \sum_{k=-N}^{N} \delta^{(0)}_{i,k} y'_u(z_k), \tag{5.45}$$

$$y_u(z_i) = \sum_{k=-N}^{N} h_k(z_i)y'_u(z_k) = \sum_{k=-N}^{N} h\delta^{(-1)}_{i,k} \frac{y'_u(z_k)}{\varphi'(z_k)}, \tag{5.46}$$

and

$$y''_u(z_i) = \sum_{k=-N}^{N} g_k(z_i)y'_u(z_k) = \sum_{k=-N}^{N} \delta^{(1)}_{i,k} \varphi'(z_i)\frac{y'_u(z_k)}{h}, \tag{5.47}$$

where

$$\delta^{(0)}_{i,k} = \begin{cases} 0, & i \neq k \\ \\ 1, & i = k \end{cases}, \tag{5.48}$$

$$\delta^{(1)}_{i,k} = \begin{cases} \frac{(-1)^{i-k}}{i-k}, & k \neq i \\ \\ 0, & k = i \end{cases}, \tag{5.49}$$

and

$$\delta^{(-1)}_{i,k} = \begin{cases} \frac{1}{2} + \int_0^{i-k} \frac{\sin(\pi t)}{\pi t}, & i \neq k \\ \\ \frac{1}{2}, & k = i \end{cases}. \tag{5.50}$$

Similarly,

$$y_v'(z_i) = \sum_{k=-N}^{N} S(k,h)(\varphi(z_i))y_v'(z_k) = \sum_{k=-N}^{N} \delta_{i,k}^{(0)} y_v'(z_k), \tag{5.51}$$

$$y_v(z_i) = \sum_{k=-N}^{N} h_k(z_i)y_v'(z_k) = \sum_{k=-N}^{N} h\delta_{i,k}^{(-1)} \frac{y_v'(z_k)}{\varphi'(z_k)}, \tag{5.52}$$

and

$$y_v''(z_i) = \sum_{k=-N}^{N} g_k(z_i)y_v'(z_k) = \sum_{k=-N}^{N} \delta_{i,k}^{(1)} \varphi'(z_i)\frac{y_v'(z_k)}{h}. \tag{5.53}$$

Thereafter, the discretized version of equations (5.35) and (5.36) will be

$$\sum_{k=-N}^{N} \Big( M_{i,k}\, y_u'(z_k) + N_{i,k}^1\, y_v'(z_k) \Big) + \Lambda_1(z_i) = F_1(z_i), \tag{5.54}$$

$$\sum_{k=-N}^{N} \Big( M_{i,k}\, y_v'(z_k) + N_{i,k}^2\, y_u'(z_k) \Big) + \Lambda_2(z_i) = F_2(z_i), \tag{5.55}$$

where

$$M_{i,k} = a(z_i)\delta_{k,i}^{(1)}\frac{\varphi'(z_i)}{h} + b(z_i)\delta_{k,i}^{(0)}, \tag{5.56}$$

$$N_{i,k}^1 = c_1(z_i)h\frac{\delta_{k,i}^{(-1)}}{\varphi'(z_k)}, \tag{5.57}$$

$$N_{i,k}^2 = c_2(z_i)h\frac{\delta_{k,i}^{(-1)}}{\varphi'(z_k)}. \tag{5.58}$$

Note that equations (5.54) and (5.55) lead to a system of ($n = 4N + 2$) equations for ($m = 4N + 10$) unknowns including the unknowns $u(a)$, $u'(a)$,

$u'(b)$, $u(b)$, $v(a)$, $v'(a)$, $v'(b)$, $v(b)$, $y_u'(z_i)$ and $y_v'(z_i), i = -N, ..., N$.

We define the $(4N + 10) \times 1$ vector $\mathbf{C}$, the union of two $(2N + 5) \times 1$ vectors called $\mathbf{C}^1$ and $\mathbf{C}^2$:

$$\mathbf{C}^1 = [C^1_{-N-2}, C^1_{-N-1}, C^1_{-N}, ...C^1_0, ..., C^1_N, C^1_{N+1}, C^1_{N+2}]^{\mathrm{T}}$$

$$= [u(a), u'(a), y_u'(x_{-N}), ..., y_u'(x_0), ...y_u'(x_N), u'(b), u(b)]^{\mathrm{T}}.$$

and

$$\mathbf{C}^2 = [C^2_{-N-2}, C^2_{-N-1}, C^2_{-N}, ...C^2_0, ..., C^2_N, C^2_{N+1}, C^2_{N+2}]^{\mathrm{T}}$$

$$= [v(a), v'(a), y_v'(x_{-N}), ..., y_v'(x_0), ...y_v'(x_N), v'(b), v(b)]^{\mathrm{T}}.$$

The eight conditions required to close the system consist of

$$\alpha^1_a C^1_{-N-2} + \beta^1_a C^1_{-N-1} = \gamma^1_a, \tag{5.59}$$

$$\alpha^1_b C^1_{N+2} + \beta^1_b C^1_{N+1} = \gamma^1_b, \tag{5.60}$$

$$\sum_{k=-N}^{N} h\delta^{(-1)}_{-N-1,k} \frac{C^1_k}{\phi'(z_k)} = 0, \tag{5.61}$$

$$\sum_{k=N}^{N} h\delta^{(-1)}_{N+1,k} \frac{C^1_k}{\phi'(z_k)} = 0, \tag{5.62}$$

and

$$\alpha^2_a C^2_{-N-2} + \beta^2_a C^2_{-N-1} = \gamma^2_a, \tag{5.63}$$

$$\alpha^2_b C^2_{N+2} + \beta^2_b C^2_{N+1} = \gamma^2_b, \tag{5.64}$$

$$\sum_{k=-N}^{N} h\delta^{(-1)}_{-N-1,k}\frac{C^2_k}{\phi'(z_k)} = 0, \tag{5.65}$$

$$\sum_{k=N}^{N} h\delta^{(-1)}_{N+1,k}\frac{C^2_k}{\phi'(z_k)} = 0. \tag{5.66}$$

Therefore, equations (5.54), (5.55), (5.59)-(5.66) constitute $(4N+10)$ equations for the $(4N+10)$ unknowns and can be represented by the matrix equation

$$\mathbf{AC} = \mathbf{F}, \tag{5.67}$$

where $\mathbf{F}$ is a $(4N+10)\times 1$ vector given by the union of two $(2N+5)\times 1$ vectors given by

$$\mathbf{F}^1 = [\gamma^1_a, \gamma^1_b, F_1(z_{-N}), ..., F_1(z_0), ..., F_1(z_N), 0, 0]^{\mathrm{T}}, \tag{5.68}$$

and

$$\mathbf{F}^2 = [\gamma^2_a, \gamma^2_b, F_2(z_{-N}), ..., F_2(z_0), ..., F_2(z_N), 0, 0]^{\mathrm{T}}, \tag{5.69}$$

and $\mathbf{A}$ is a $(4N+10)\times(4N+10)$ matrix given by

$$A = \left[\begin{array}{c|c} A^1 & A^2 \\ \hline A^3 & A^4 \end{array}\right],$$

where

$$\mathbf{A}^1 = \begin{pmatrix} B_1 \\ B_2 \\ \mathbf{B} \\ B_3 \\ B_4 \end{pmatrix}, \tag{5.70}$$

and $B_1$, $B_2$, $B_3$ and $B_4$ are $1 \times (2N+5)$ matrices given by

$$B_1 = [\alpha_a^1, \beta_a^1, 0, ..., 0],$$

$$B_2 = [0, 0, ..., \beta_b^1, \alpha_b^1],$$

$$B_3 = [0, 0, \frac{h\delta^{(-1)}_{-N-1,-N}}{\phi'(z_k)}, ..., \frac{h\delta^{(-1)}_{-N-1,N}}{\phi'(z_k)}, 0, 0],$$

$$B_4 = [0, 0, \frac{h\delta^{(-1)}_{N+1,-N}}{\phi'(z_k)}, ..., \frac{h\delta^{(-1)}_{N+1,N}}{\phi'(z_k)}, 0, 0],$$

and $\mathbf{B}$ is an $(2N+1) \times (2N+5)$ matrix given by

$$\mathbf{B} = [\eta_2^T, \eta_1^T, \mathbf{M}, \eta_4^T, \eta_3^T],$$

where $\mathbf{M}$ is the $(2N+1) \times (2N+1)$ matrix represented by equation (5.56). Please note that $\eta_i^T$, is the transpose of $\eta_i$.

Matrix $A^4$ is equal to matrix $A^1$. Matrix $A^2$ is a $(2N+5) \times (2N+5)$

matrix given by

$$\mathbf{A}^2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B*} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \tag{5.71}$$

where

$$\mathbf{B*} = [\zeta_2^T, \zeta_1^T, \mathbf{N}^1, \zeta_4^T, \zeta_3^T],$$

and $\mathbf{0}$=[0, ..., 0] is a $(1) \times (2N+5)$ vector of zeros. Please note that $\zeta_i^T$, is the transpose of $\zeta_i$. Since $c_2(z) = -c_1(z)$,

$$\mathbf{A}^3 = -\mathbf{A}^2.$$

Once equation (5.67) is solved, the coefficients are used to determine the unknown functions $y_u(z)$ and $y_v(z)$ and their first and second derivatives at the sinc nodes using equations and (5.45)-(5.47) and (5.51)-(5.53). The original unknowns, $u(z)$ and $v(z)$ are then determined from the equations given in (5.31) and (5.32). To calculate $U(z)$ and $V(z)$ we need to apply equations given in (4.15).

# Chapter 6

# Numerical Illustrations

## 6.1 Constant Eddy Viscosity

In this section we investigate the accuracy of our Sinc-Collocation method in both complex velocity and coupled systems with constant eddy viscosity. To provide reliable comparisons, all the examples, parameters and variables are set exactly the same as those used in [71], [35], and [36].

Since the governing equations and variables were nondimensionalized, the only operative constants in (4.28)-(4.30) are $\kappa = \frac{D_0}{D_E} = \frac{100}{20} = 5, \sigma = \frac{A^*{}_v(D_0)}{(k_f D_0)} = 0.1$, and $\chi = 45^\circ$ [71]. As well, the following nominal values: $f = 0.0001$ s$^{-1}$ (appropriate to mid-northern latitudes), sea water density $\rho = 1 \times 10^3$ kgm$^{-3}$, and air density $\rho_{air} = 1.25$ kgm$^{-3}$ are considered. The surface wind stress (in ms$^{-1}$) is given by

$$\tau_w = C_D \rho_{air} {W_w}^2 \tag{6.1}$$

where the dimensionless parameter $C_D \approx 0.0012$ for $W_w < 12$ ms$^{-1}$, thereafter increasing linearly to about 0.0025 at gale force winds ($W_w \approx 30$ ms$^{-1}$) [71]. The linear slip bottom stress coefficient $k_f$ is set at 0.002 ms$^{-1}$. If the wind is not fetch-limited and the sea state is fully developed, $A^*{}_v(0)$ in units of m$^2$s$^{-1}$ is given by [71]

$$A^*{}_v(0) \approx 0.304 \times 10^{-4} {W_w}^3 \tag{6.2}$$

Together with the parameters and relationships above, the constant eddy viscosity is chosen to be

$$A^*{}_v(z^*) \equiv 0.02 \;\; m^2 s^{-1} \tag{6.3}$$

with $\tau_w = \frac{\sqrt(2)}{10} = 0.1414 \quad Nm^{-2}$.

To exhibit the accuracy of the Sinc-Collocation method we compare the numerical results to the exact solutions whenever possible. Likewise, to show the excellence of our Sinc-Collocation approach over Sinc-Galerkin approach we provide a comparison between our method's error and those in references [71], [35], and [36].

In the case of constant eddy viscosity, the exact solution is available and given by $W^*(z^*) = U_0[U(z) + iV(z)]$ where $U(z)$ and $V(z)$ are respectively represented by

$$U(z) = \mathcal{R}(W_c(z))\cos(\chi) - \mathcal{I}(W_c(z))\sin(\chi), \tag{6.4}$$

and

$$V(z) = \mathcal{R}(W_c(z))\sin(\chi) + \mathcal{I}(W_c(z))\cos(\chi). \tag{6.5}$$

Here $\mathcal{R}(W_c(z))$ and $\mathcal{I}(W_c(z))$ refers to the real and imaginary parts of $W_c(z)$, respectively, while

$$W_c(z) = \frac{\kappa(1-i)\sigma\cosh(\kappa(1-i)(1-z)) + \sinh(\kappa(1-i)(1-z))}{(1-i)[\cosh(\kappa(1-i)) + \kappa(1-i)\sigma\sinh(\kappa(1-i))]}. \tag{6.6}$$

The results of the Sinc-Collocation approach shown by $U_s(z_j)$ and $V_s(z_j)$ were compared with the exact solutions, $U(z_j)$ and $V(z_j)$, at the sinc grid points $\mathcal{S}$ with the mesh size of

$$h = \frac{\log(\pi d\gamma N/\beta)}{\gamma N}$$

where $d, \gamma$, and $\beta$ are equal to $\frac{\pi}{4}$, 2, and $\frac{\pi}{2}$ respectively.

In order to provide dimensional representation of the velocities, we need to multiply the results by the natural velocity scale $U_0$.

To demonstrate the accuracy of the method, we define the maximum absolute errors by

$$\|E_U\| = \max_{-N-2\leq j\leq N+2}\{U_0|U_s(z_j) - U(z_j)|\},$$

$$\|E_V\| = \max_{-N-2\leq j\leq N+2}\{U_0|V_s(z_j) - V(z_j)|\},$$

and

$$\|E_W\|= max\{\|E_U\|, \|E_V\|\}. \tag{6.7}$$

where the units are ms$^{-1}$.

To illustrate the computation cost of our method, we report computation times by CPU (in seconds) which are calculated using the Mathematica command “Timing”. We run our code in a SONY VAIO (model name: VGN-Z690C) with a 64-bit operating system and an Intel(R) dual core processor. The command “Timing” includes only CPU time spent in the evaluation of an expression in the Mathematica kernel. It does not include the time spent in formatting or printing of the result. The command “Timing” may give different results on different occasions within a session which is due to internal system caches. To prevent this issue we cleared the system cache using the command “ClearSystemCache”.

**Example 1.a**.(seabed linear stress condition in the complex velocity system) For the purpose of keeping the parameters and variables identical to references [71] and [35], we choose $\chi = 45°$ and the linear stress condition at the seabed, $\sigma = \frac{A^*{}_v(D_0)}{(k_f D_0)} = 0.1$. In this example we solve a discrete system of size $(2N + 5) \times (2N + 5)$ given by (5.21). To demonstrate the numerical convergence of the method we consider N = 4, 8, 16, 32, and 64. The errors and CPU times are listed in Table 6.1. Figure 6.1 presents Ekman spiral of the case discussed here. For N=64, we obviously see a close agreement between the exact solution and the approximate solution. Table 6.2, provides a comparison between the errors and CPU times of the Sinc-Collocation method and those in [71] and [35] which are based on the Sinc-Galerkin scheme. $E_W, E_2$ and $E_3$ convey the maximum errors of our method, those in [71] and [35], respectively. While the current method outperforms the other methods, the CPU times taken for the simulated cases are similar to those in [71]. This is expected since the sparsness of the matrix in the final linear system is very similar to that of the other methods. Hence the current method is clearly more efficient.

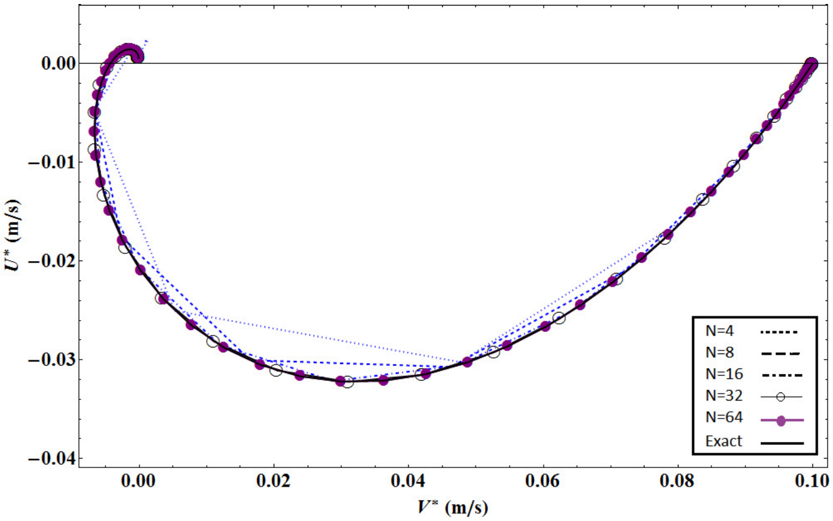


Figure 6.1: The Sinc-Collocation Ekman Spiral projection of Example 1.a for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\, m, D_E = 20\, m$.

**Example 1.b**. (seabed linear stress condition in the coupled system)

We repeat Example 1.a using the coupled system to check if it gives us better approximations than the complex system. In this example we solve a dis-

Table 6.1: Errors of Example 1.a (constant eddy viscosity in the complex system) with $\sigma = 0.1, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | CPU (s) | $\|E_U\|$ | $\|E_V\|$ | $\|E_W\|$ |
|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $2.9852 \times 10^{-3}$ | $3.4708 \times 10^{-3}$ | $3.4708 \times 10^{-3}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.2634 \times 10^{-4}$ | $8.4080 \times 10^{-5}$ | $1.2634 \times 10^{-4}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4903 \times 10^{-6}$ | $1.2267 \times 10^{-6}$ | $2.4903 \times 10^{-6}$ |
| 32 | 69 | 0.0720 | 0.125 | $2.9558 \times 10^{-8}$ | $1.4260 \times 10^{-8}$ | $2.9558 \times 10^{-8}$ |
| 64 | 133 | 0.0414 | 0.156 | $1.2276 \times 10^{-10}$ | $1.817 \times 10^{-10}$ | $1.817 \times 10^{-10}$ |

Table 6.2: A comparison between the errors in Example 1.a (in the complex system) and those in papers [35, 71], with $\sigma = 0.1, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | CPU (s) | $\|E_W\|$ | CPU (s) | $\|E_2\|$ | $\|E_3\|$ |
|---|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $3.4708 \times 10^{-3}$ | 0.01 | $1.10 \times 10^{-3}$ | $5.377 \times 10^{-2}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.2634 \times 10^{-4}$ | 0.01 | $2.50 \times 10^{-4}$ | $4.571 \times 10^{-2}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4903 \times 10^{-6}$ | 0.03 | $2.76 \times 10^{-5}$ | $1.861 \times 10^{-2}$ |
| 32 | 69 | 0.0720 | 0.125 | $2.9558 \times 10^{-8}$ | 0.15 | $8.99 \times 10^{-7}$ | $8.189 \times 10^{-3}$ |
| 64 | 133 | 0.0414 | 0.156 | $1.817 \times 10^{-10}$ | 1.01 | $5.78 \times 10^{-9}$ | $7.13 \times 10^{-4}$ |

Table 6.3: Errors of Example 1.b (constant eddy viscosity in the coupled system) with $\sigma = 0.1, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | CPU (s) | $\|E_U\|$ | $\|E_V\|$ | $\|E_W\|$ |
|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $2.9852 \times 10^{-3}$ | $3.4708 \times 10^{-3}$ | $3.4708 \times 10^{-3}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.2634 \times 10^{-4}$ | $8.4080 \times 10^{-5}$ | $1.2634 \times 10^{-4}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4903 \times 10^{-6}$ | $1.2268 \times 10^{-6}$ | $2.4903 \times 10^{-6}$ |
| 32 | 69 | 0.0720 | 0.078 | $2.9558 \times 10^{-8}$ | $1.4260 \times 10^{-8}$ | $2.9558 \times 10^{-8}$ |
| 64 | 133 | 0.0414 | 0.109 | $7.2213 \times 10^{-11}$ | $4.8278 \times 10^{-11}$ | $7.2213 \times 10^{-11}$ |

Table 6.4: A comparison between the errors in Example 1.b (in the coupled system) and those in paper [36], with $\sigma = 0.1, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | $\|E_W\|$ | $\|E_3\|$ |
|---|---|---|---|---|
| 4 | 13 | 0.3163 | $3.4708 \times 10^{-3}$ | $5.377 \times 10^{-2}$ |
| 8 | 21 | 0.2015 | $1.2634 \times 10^{-4}$ | $4.571 \times 10^{-2}$ |
| 16 | 37 | 0.1224 | $2.4903 \times 10^{-6}$ | $1.861 \times 10^{-2}$ |
| 32 | 69 | 0.0720 | $2.9558 \times 10^{-8}$ | $8.189 \times 10^{-3}$ |
| 64 | 133 | 0.0414 | $7.2213 \times 10^{-11}$ | $7.13 \times 10^{-4}$ |

crete system of size $(4N+10) \times (4N+10)$ given by (5.67). Table 6.3 exhibits the errors and CPU times of the Sinc-Collocation approach applied to the coupled system. In addition, a comparison between the errors of our method and those in [36] is provided in Table 6.4. The corresponding Ekman spiral to Example 1.b is depicted in Figure 6.2. Comparing the errors in Tables 6.1 and 6.3, shows that the only difference between errors of the complex system and the coupled system happens at N = 64. The Sinc-Collocation approach in the coupled system provides more accurate approximation than that in the complex system for N = 64.

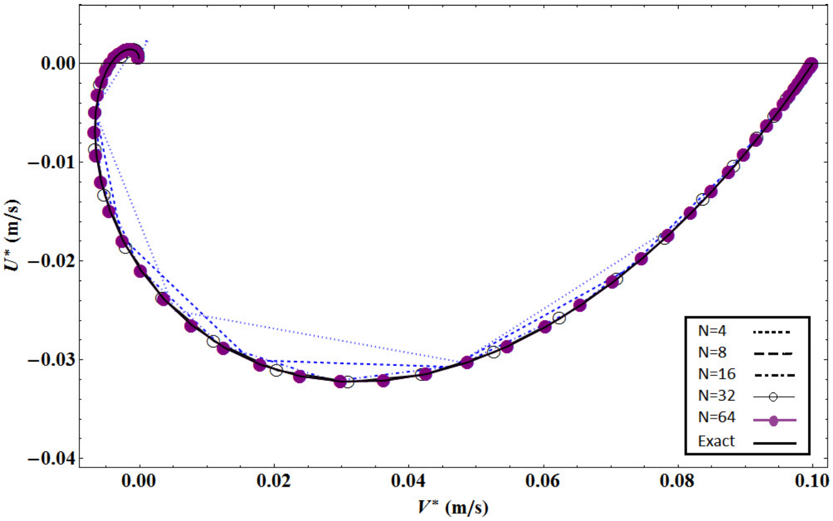


Figure 6.2: The Sinc-Collocation Ekman Spiral projection of Example 1.b for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\, m, D_E = 20\, m$.

**Example 2.a**. (No-slip condition at the seabed in the complex system)

In this example we set $\sigma = 0$. All other parameters are similar to references [71] and [35] and those carried out in Example 1.a. Here, we solve the discrete system given by (5.21) for approximate solutions $U_s(z)$ and $V_s(z)$. The errors and CPU times for different values of N are listed in Table 6.5 and a close agreement to those in Example 1.a is found. The horizontal projection of the Ekman spiral for different values of N against the exact solution are portrayed in Figure 6.3. Likewise, Table 6.6 provides a comparison between the maximum errors and CPU times of our method, those in [71] and [35], respectively. While the current method outperforms the other methods, the CPU times taken for the simulated cases are similar to those in [71]. This is expected since the sparsness of the matrix in the final linear system is very

Table 6.5: Errors of Example 2.a (constant eddy viscosity in the complex system) with $\sigma = 0, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | CPU (s) | $\|E_U\|$ | $\|E_V\|$ | $\|E_W\|$ |
|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $3.0613 \times 10^{-3}$ | $3.3831 \times 10^{-3}$ | $3.3831 \times 10^{-3}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.25 \times 10^{-4}$ | $8.4230 \times 10^{-5}$ | $1.25 \times 10^{-4}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4824 \times 10^{-6}$ | $1.2312 \times 10^{-6}$ | $2.4824 \times 10^{-6}$ |
| 32 | 69 | 0.0720 | 0.125 | $2.9460 \times 10^{-8}$ | $1.4316 \times 10^{-8}$ | $2.9460 \times 10^{-8}$ |
| 64 | 133 | 0.0414 | 0.156 | $8.2568 \times 10^{-11}$ | $8.3657 \times 10^{-11}$ | $8.3657 \times 10^{-11}$ |

similar to that of the other methods. Hence the current method is clearly more efficient.

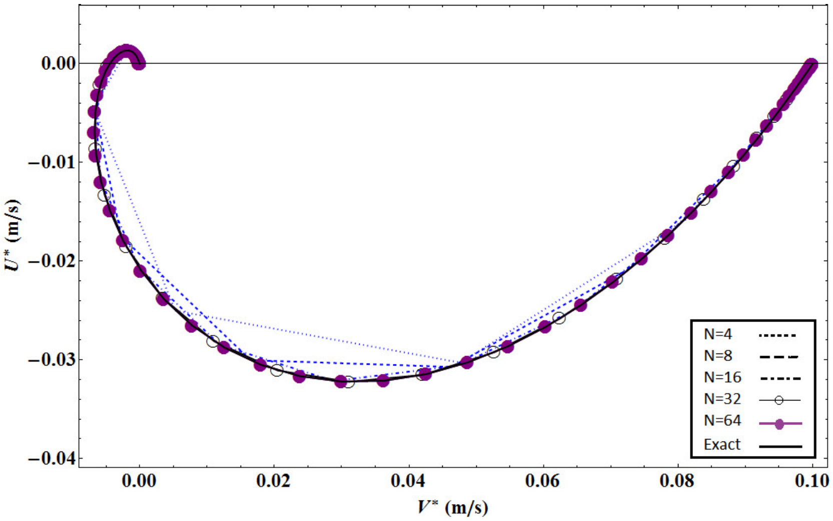


Figure 6.3: The Sinc-Collocation Ekman Spiral projection of Example 2.a (in the complex system) for different values of N against the exact solution with $\sigma = 0, \chi = 45, \kappa = 5, D_0 = 100\,m, D_E = 20\,m$.

**Example 2.b.**(No-slip condition at the seabed in the coupled system)

Here, we repeat Example 2.a in the coupled system. Therefore, we solve the system given by (5.67). Table 6.7 exhibits the errors and CPU times of

Table 6.6: A comparison between the errors in Example 2.a (in the complex system) and those in [35, 71], with $\sigma = 0, \chi = 45°, \kappa = 5, D_0 = 100\, m$ and $D_E = 20\, m$.

| N | m | h | CPU (s) | $\|E_W\|$ | CPU (s) | $\|E_2\|$ | $\|E_3\|$ |
|---|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $3.3831 \times 10^{-3}$ | 0.01 | $1.10 \times 10^{-3}$ | $5.3322 \times 10^{-2}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.25 \times 10^{-4}$ | 0.01 | $2.48 \times 10^{-4}$ | $4.5478 \times 10^{-2}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4824 \times 10^{-6}$ | 0.03 | $2.75 \times 10^{-5}$ | $1.8543 \times 10^{-2}$ |
| 32 | 69 | 0.0720 | 0.125 | $2.9460 \times 10^{-8}$ | 0.16 | $8.96 \times 10^{-7}$ | $8.1680 \times 10^{-3}$ |
| 64 | 133 | 0.0414 | 0.156 | $8.3657 \times 10^{-11}$ | 1.07 | $5.76 \times 10^{-9}$ | $7.1 \times 10^{-4}$ |

Table 6.7: Errors of Example 2.b (constant eddy viscosity in the coupled system) with $\sigma = 0, \chi = 45°, \kappa = 5, D_0 = 100\, m$ and $D_E = 20\, m$.

| N | m | h | CPU (s) | $\|E_U\|$ | $\|E_V\|$ | $\|E_W\|$ |
|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $3.0613 \times 10^{-3}$ | $3.3831 \times 10^{-3}$ | $3.3831 \times 10^{-3}$ |
| 8 | 21 | 0.2015 | 0.016 | $1.25 \times 10^{-4}$ | $8.4230 \times 10^{-5}$ | $1.25 \times 10^{-4}$ |
| 16 | 37 | 0.1224 | 0.016 | $2.4825 \times 10^{-6}$ | $1.2312 \times 10^{-6}$ | $2.4825 \times 10^{-6}$ |
| 32 | 69 | 0.0720 | 0.078 | $2.9460 \times 10^{-8}$ | $1.4316 \times 10^{-8}$ | $2.9460 \times 10^{-8}$ |
| 64 | 133 | 0.0414 | 0.109 | $9.1851 \times 10^{-11}$ | $4.4308 \times 10^{-11}$ | $9.1851 \times 10^{-11}$ |

the Sinc-Collocation approach applied to the coupled system. In addition, a comparison between the errors of our method and those in [36] is provided in Table 6.8. The corresponding Ekman spiral to Example 2.b is depicted in Figure 6.4. Comparing the errors in Tables 6.5 and 6.7, shows that the only difference between errors of the complex system and the coupled system occurs at N = 64. The Sinc-Collocation approach in the coupled system provides more accurate approximation than that in the complex system for N = 64.

Table 6.8: A comparison between the errors in Example 2.b (in the coupled system) and those in [36], with $\sigma = 0, \chi = 45°, \kappa = 5, D_0 = 100\,m$ and $D_E = 20\,m$.

| N | m | h | $\|E_W\|$ | $\|E_3\|$ |
|---|---|---|---|---|
| 4 | 13 | 0.3163 | $3.3831 \times 10^{-3}$ | $5.3322 \times 10^{-2}$ |
| 8 | 21 | 0.2015 | $1.25 \times 10^{-4}$ | $4.5478 \times 10^{-2}$ |
| 16 | 37 | 0.1224 | $2.4824 \times 10^{-6}$ | $1.8543 \times 10^{-2}$ |
| 32 | 69 | 0.0720 | $2.9460 \times 10^{-8}$ | $8.1680 \times 10^{-3}$ |
| 64 | 133 | 0.0414 | $9.1851 \times 10^{-11}$ | $7.1 \times 10^{-4}$ |

## 6.2 Variable Eddy Viscosity

In the real world the eddy viscosity is a depth- and time-dependent variable. But the model we worked on, specifically does study the depth-dependent eddy viscosity. There is a specific model problem in [35] containing the time-dependent eddy viscosity where as $t \to \infty$, the eddy viscosity can be considered as a constant. Since the latter case is similar to those in Examples 1.a to 2.b, we consider it in our work.

In seas of shallow to intermediate depth, the eddy viscosity has the maximum values of ${A^*}_v(z^*)$ at the intermediate depths and the minimum values near the surface and seabed. But in deeper seas, it is expected that ${A^*}_v(z^*)$ has the maximum values near the surface and its value decreases going towards the seabed. The latter case is illustrated by

$${A^*}_v(z^*) = 0.02[1 - (0.0075)z^*]^2, \quad 0 < z^* < D_0 = 100\ m, \tag{6.8}$$

which decreases quadratically from the value of ${A^*}_v(0) = 0.02\ \mathrm{m}^2\ \mathrm{s}^{-1}$ to the

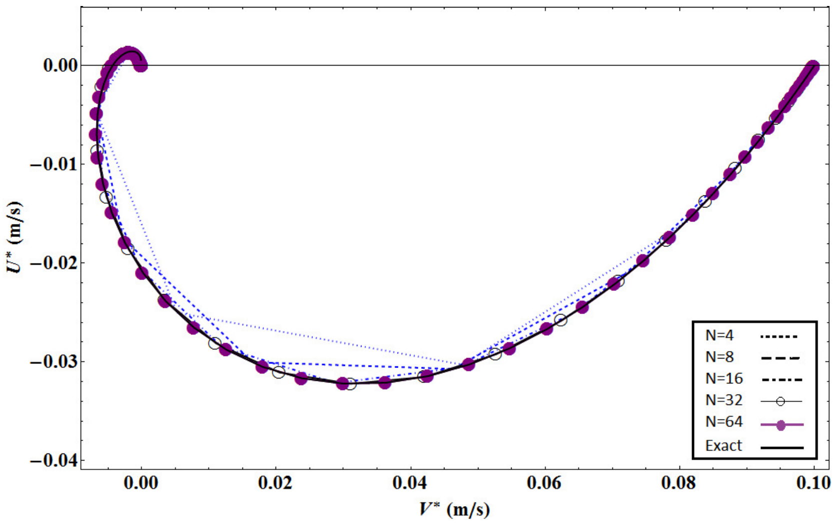


Figure 6.4: The Sinc-Collocation Ekman Spiral projection of Example 2.b (in the coupled system) for different values of N against the exact solution with $\sigma = 0, \chi = 45, \kappa = 5, D_0 = 100\,m, D_E = 20\,m$.

minimum value of $A^*{}_v(D_0) = 0.00125$ m$^2$ s$^{-1}$. In Figure 6.5 we portray the decreasing eddy viscosity against the constant eddy viscosity.

The eddy viscosity in the first case follows a quadratic model given by

$$A^*{}_v(z^*) = 0.02[1 + (0.12)z^*(1 - (0.01)z^*)], \quad 0 < z^* < D_0 = 100\ m. \quad (6.9)$$

increasing from the initial value of $A^*{}_v(0) = 0.02$ m$^2$s$^{-1}$ to the peak value of 0.08 and then decreasing to $A^*{}_v(D_0) = 0.02$ m$^2$ s$^{-1}$. Figure 6.6 compares the quadratic eddy viscosity with the constant eddy viscosity.

In the following part, we solve the presented model in both cases of variable eddy viscosity by the complex velocity and the coupled discrete systems.

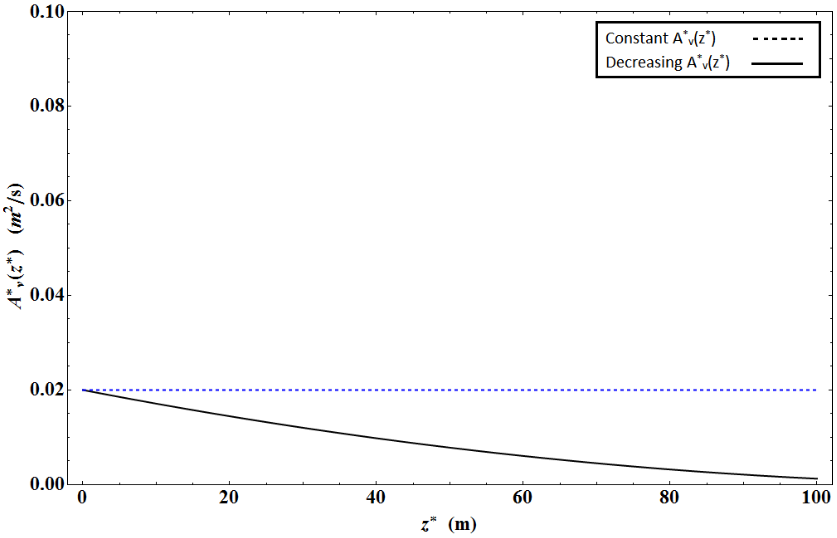


Figure 6.5: Decreasing eddy viscosity function $A_v^*(z^*) = 0.02(1 - 0.0075z^*)^2$ and constant eddy viscosity $A_v^*(z^*) = 0.02 \ (m^2/s)$.

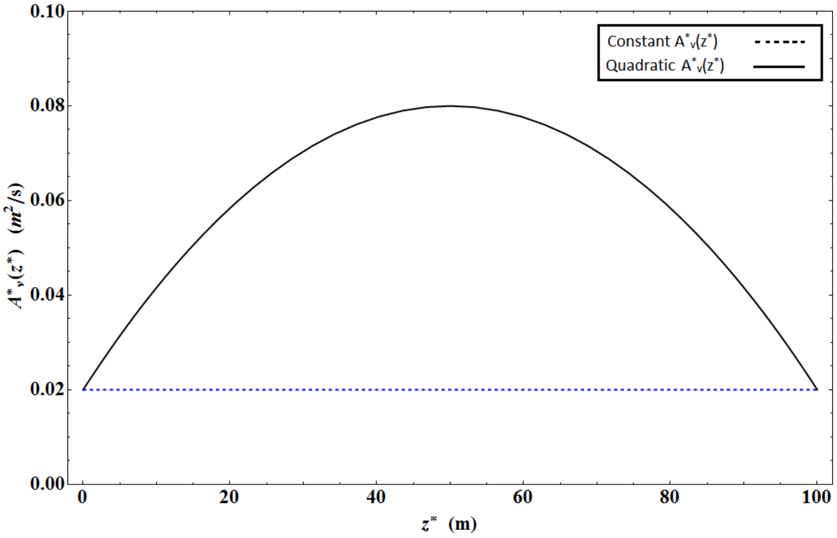


Figure 6.6: Quadratic eddy viscosity function $A_v^*(z^*) = 0.02[1 + (0.12)z^*(1 - (0.01)z^*)]^2$ and constant eddy viscosity $A_v^*(z^*) = 0.02 \ (m^2/s)$.

**Example 3.a.**(The decreasing eddy viscosity in the complex velocity system) In this example we find the approximate solutions $U_s(z)$ and $V_s(z)$ via the complex velocity discrete system (5.21), when the variable eddy viscosity is given by (6.8). The parameters are chosen identical to those in references [71] and [35]. Hence, $D_0 = 100$ m, $\sigma = 0.1$, $\chi = 45°$, and $\kappa = 5$.

Since there is no closed form solution for the current case, we present the Ekman spiral projection of decreasing eddy viscosity in the complex velocity system against that of constant eddy viscosity for different values of N, in Figure 6.7.

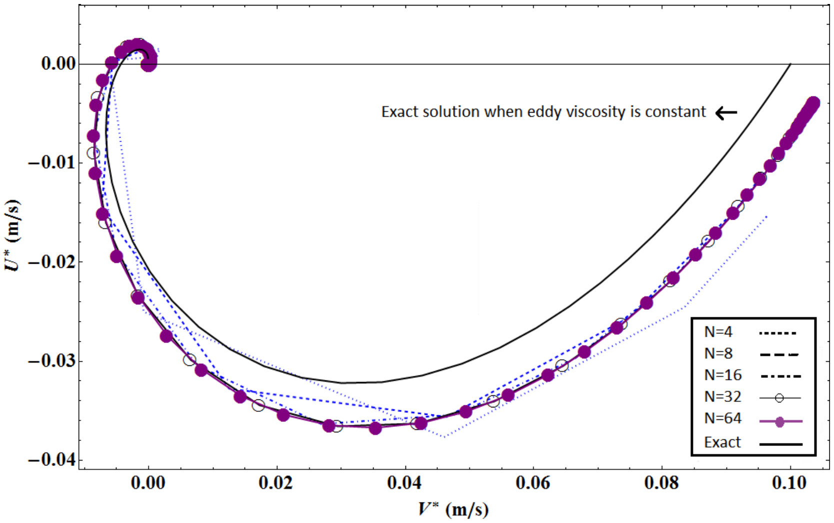


Figure 6.7: The Sinc-Collocation Ekman Spiral projection of Example 3.a (in the complex system) for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\,m, D_E = 20\,m$.

**Example 3.b.** (The decreasing eddy viscosity in the coupled system)

We applied our approach to the same model problem in Example 3.a but in the coupled system. Since there is no closed form solution of this case, we present the results by the Ekman spiral projection of the decreasing eddy viscosity in the coupled system against that of constant eddy viscosity for different values of N, in Figure 6.8.

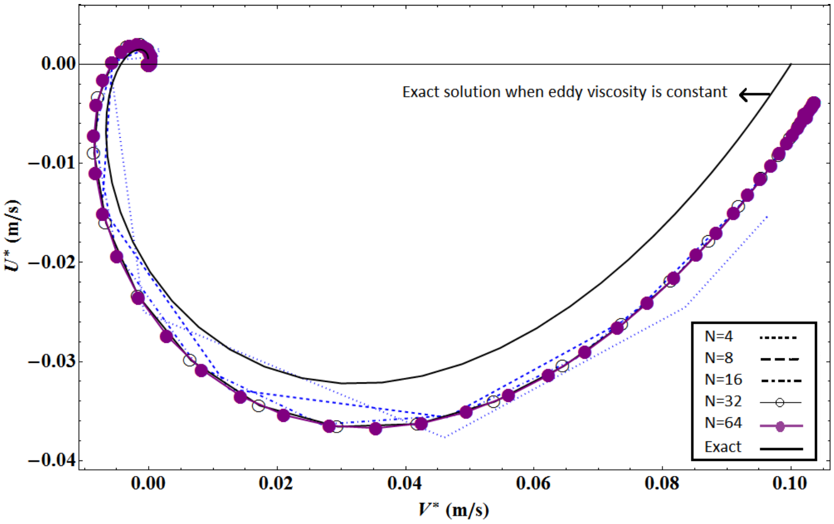


Figure 6.8: The Sinc-Collocation Ekman Spiral projection of Example 3.b (in the coupled system) for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\, m, D_E = 20\, m$.

**Example 4.a**. (The quadratic eddy viscosity in the complex velocity system) In this example, the approximate solutions $U_s(z)$ and $V_s(z)$ are demonstrated by the complex velocity discrete system while the eddy viscosity is given by (6.9). All the parameters are identical to those carried out in Example 1.a. The exact solution of this model problem is not appropriate. Therefore to discuss the results, we portray the Ekman spiral projection of quadratic eddy viscosity in the complex system against that of constant eddy viscosity for

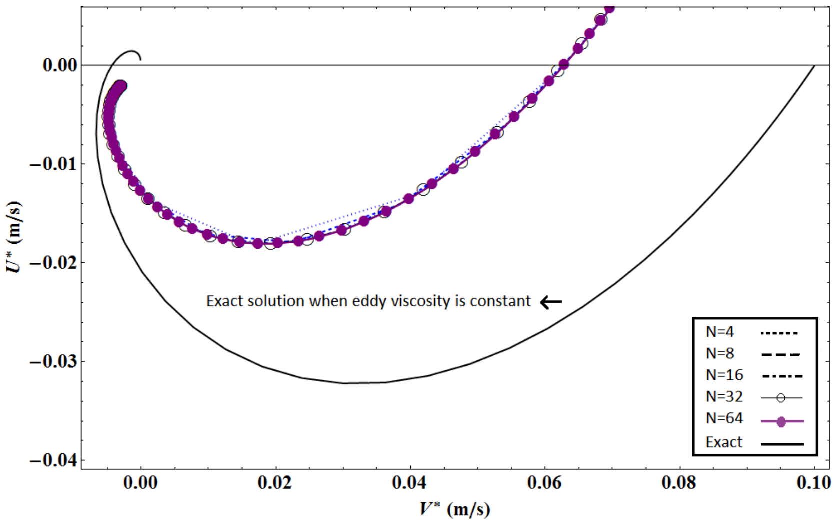


Figure 6.9: The Sinc-Collocation Ekman Spiral projection of Example 4.a (in the complex system) for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\,m, D_E = 20\,m$.

different values of N by Figure 6.9.

**Example 4.b**. (The quadratic eddy viscosity in the coupled system)

The same model problem as Example 4.a is investigated in the coupled system. Figure 6.10 shows the results by the help of the Ekman spiral projection of quadratic eddy viscosity in the coupled system against that of constant eddy viscosity for different values of N.

**Example 5**. (A steady-state problem in the complex system)

As discussed earlier, eddy viscosity is a time- and depth-dependent variable. Realistic oceanography problems are those in which eddy viscosity is a function of depth and time. Field studies show that the value of the eddy viscosity

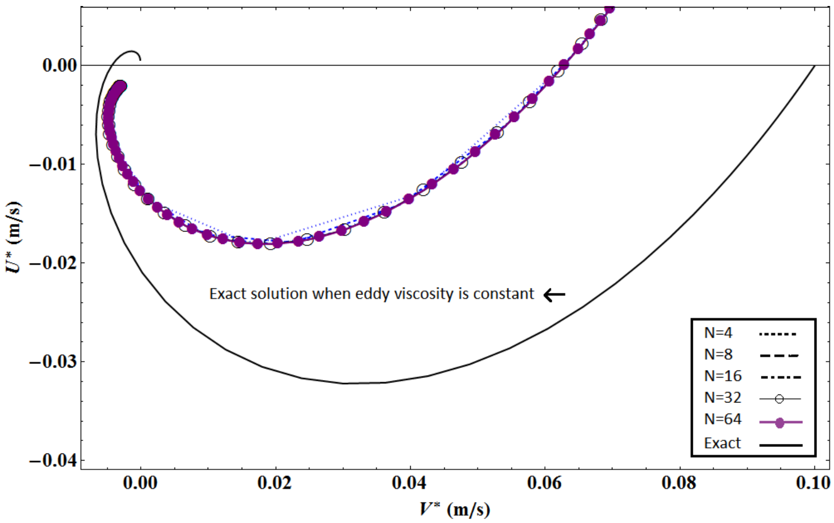


Figure 6.10: The Sinc-Collocation Ekman Spiral projection of Example 4.b (in the coupled system) for different values of N against the exact solution with $\sigma = 0.1, \chi = 45, \kappa = 5, D_0 = 100\,m, D_E = 20\,m$.

near the surface is dependent on the wind stress which relies on time. Therefore, in shallow seas ($D_0 < 100\,m$), the eddy viscosity is assumed dependent of time but independent of depth. Here, we study an interesting example of this case found in [35].

Assume the nondimensional time-dependent eddy viscosity

$$A_v(t) = 4 - 3e^{-t}.$$

At the steady-state condition ($t \to \infty$), it will be equivalent to $A_\infty \equiv 4$. Then consider the steady-state boundary value problem

$$A_\infty \frac{d^2w(z)}{dz^2} + 2\kappa^2 i w(z) = -2\kappa^3 i \left(\frac{1-z}{A_\infty}\right) e^{i\chi} \tag{6.10}$$

with time-independent boundary conditions

$$\frac{dw(0)}{dz} = 0, \tag{6.11}$$

$$w(1) = 0. \tag{6.12}$$

and the no-slip boundary condition $\sigma = 0$.

The exact solution of this problem is $W(z) = U_0(U(z) + iV(z))$, where U(z) and V(z) are given by

$$U(z) = R(W_c(z))\cos(\chi) - I(W_c(z))\sin(\chi),$$

$$V(z) = R(W_c(z))\sin(\chi) - I(W_c(z))\cos(\chi),$$

and

$$W_c(z) = \left(\frac{1+i}{2}\right)\frac{\sinh\left((1-i)\kappa(1-z)\sqrt{\frac{1}{A_\infty}}\right)}{\sqrt{A_\infty}\cosh\left((1-i)\kappa\sqrt{\frac{1}{A_\infty}}\right)}. \tag{6.13}$$

This example is similar to Example 2.a. So we solved the problem by the complex dicrete system in (5.21). The results comparing to the exact solution are depicted in Table 6.9. Figure 6.11, displays the Ekman spiral projection of the steady-state problem for N = 4, 8, 16, 32, and 64 against the exact solution. In Table 6.10, we compare our results with those in [35].

To illustrate that the errors of our method exponentially decay, we present the logarithmic plots of the maximum errors of Examples 1.a, 2.a and 5. Ta-

Table 6.9: Errors of Example 5 (the steady-state problem in the complex system) with $\sigma = 0, \chi = 45°, \kappa = 3.14, D_0 = 60\, m$ and $D_E = 19\, m$.

| N | m | h | CPU (s) | $\|E_U\|$ | $\|E_V\|$ | $\|E_W\|$ |
|---|---|---|---|---|---|---|
| 4 | 13 | 0.3163 | 0.015 | $9.6571 \times 10^{-6}$ | $3.4867 \times 10^{-6}$ | $9.6571 \times 10^{-6}$ |
| 8 | 21 | 0.2015 | 0.015 | $4.5835 \times 10^{-8}$ | $9.9910 \times 10^{-8}$ | $9.9910 \times 10^{-8}$ |
| 16 | 37 | 0.1224 | 0.016 | $5.833 \times 10^{-9}$ | $2.1325 \times 10^{-9}$ | $5.833 \times 10^{-9}$ |
| 32 | 69 | 0.0720 | 0.093 | $6.9410 \times 10^{-11}$ | $2.5413 \times 10^{-11}$ | $6.9410 \times 10^{-11}$ |
| 64 | 133 | 0.0414 | 0.141 | $2.4219 \times 10^{-13}$ | $3.0335 \times 10^{-13}$ | $3.0335 \times 10^{-13}$ |

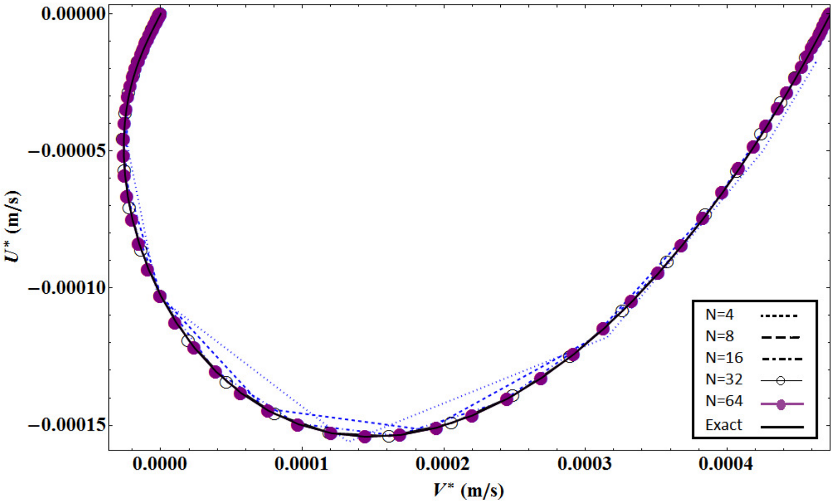


Figure 6.11: The Sinc-Collocation Ekman Spiral projection of Example 5 for different values of N against the exact solution with $\sigma = 0, \chi = 45, \kappa = 3.14, D_0 = 60\, m, D_E = 19\, m$.

Table 6.10: A comparison between the errors in Example 5 (in the complex system) and those in [35], with $\sigma = 0, \chi = 45°, \kappa = 3.14, D_0 = 60\, m$ and $D_E = 19\, m$.

| N | m | h | $\|E_W\|$ | $\|E_3\|$ |
|---|---|---|---|---|
| 4 | 13 | 0.3163 | $9.6571 \times 10^{-6}$ | $2.1261 \times 10^{-1}$ |
| 8 | 21 | 0.2015 | $9.9910 \times 10^{-8}$ | $2.7372 \times 10^{-1}$ |
| 16 | 37 | 0.1224 | $5.833 \times 10^{-9}$ | $8.6065 \times 10^{-2}$ |
| 32 | 69 | 0.0720 | $6.9410 \times 10^{-11}$ | $2.2573 \times 10^{-2}$ |

bles 6.12 and 6.13 represent the $log_{10}(E_w)$ against the number of collocation nodes regarding the maximum errors observed in Examples 1.a and 2.a. Table 6.14 demonstrates the logarithmic plot of the errors observed in Example 5.

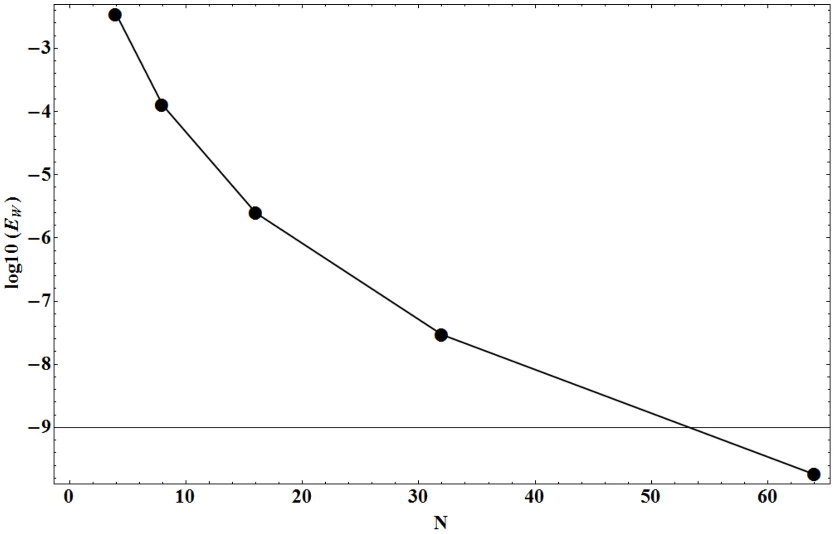


Figure 6.12: The logarithmic plot of the errors observed in Example 1.a

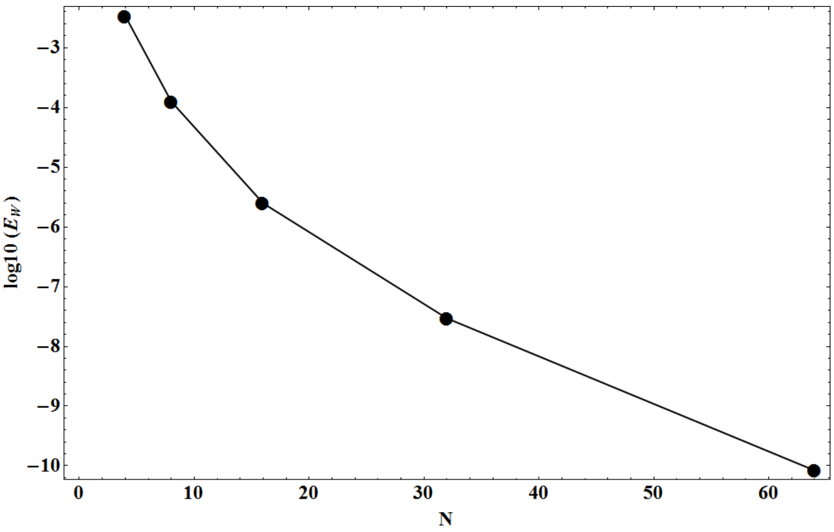


Figure 6.13: The logarithmic plot of the errors observed in Example 2.a

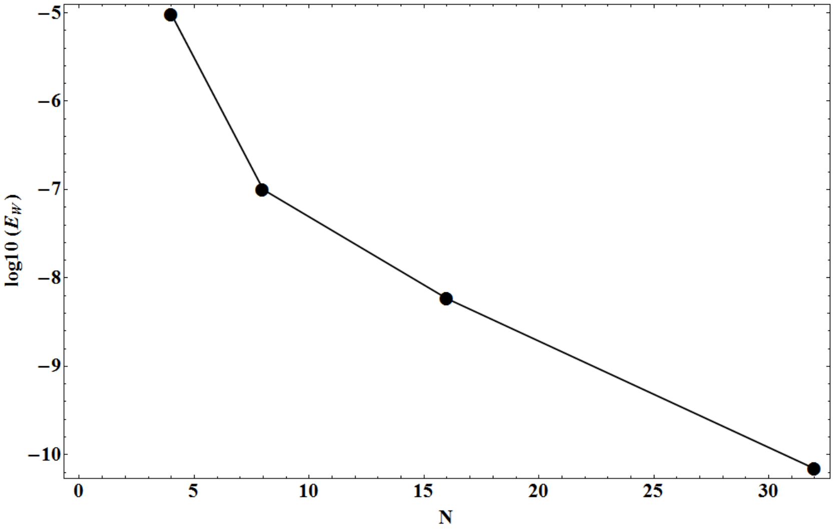


Figure 6.14: The logarithmic plot of the errors observed in Example 5

# Chapter 7

# Conclusions

During the last century significant efforts have been made to create the most elaborate oceanography models and find appropriate solutions for them. Different methods have been applied in this area to obtain the most accurate approximations. Since in most cases deriving an analytical solution is computationally expensive or completely impossible, we rely on numerical methods to approximate the solutions. Among all numerical methods, we highly recommend Sinc-based methods because they are adaptable in handling problems with singularities which frequently occur in fluid mechanics. In addition, their errors have an exponential convergence rate tending to zero when the number of collocation points increases. To the best of our knowledge, the Sinc-Galerkin approaches have been applied in this area but the Sinc-Collocation approach has not still been utilized. In this thesis, we applied a Sinc-Collocation technique to numerically approximate the solution of an oceanography model originated in [71]. The validity of our approach is demonstrated by solving illustrative examples found in [35, 36, 71] and comparing the results with the exact solutions and those in prior studies. Our results show that the presented Sinc-Collocation approach is computationally less expensive than the Sinc-Galerkin approach presented in [71] and more accurate than prior studies. Hence, we would claim that our Sinc-Collocation approach is very promising in oceanographic problems. In closing, we propose the current method as an alternative to other methods which have been used thus far.

Future research may include an investigation of the convergence rate and detailed error analysis to obtain efficient and reliable error indicators. Since the real world oceanography problems are given by partial differential equations, and time is an important factor specially when it comes to optimization, one may investigate the model in which eddy viscosity is a depth- and time-dependent variable and solve the problem utilizing the approach presented here.

# Appendix A

# The Mathematica Source Code

## A.1 The Mathematica Source Code for the Complex System

Here the code for solving Example 1.a in the complex system is attached. This part is dedicated to the definition of essential functions and assignment of the desired values to variables.

```
$MinPrecision = 16;
$MaxPrecision = 64;
NDEC:= 32;
Ns:= 64;
χ:= π/4;
κ:= 5;
DZ:= 100;
```

```
DE:= 20;

σ:= 0;

ww:= 9;

CD:= 0.0012;

τw:= 0.1414;

ρ:= 1000;

Anull:= 0.02;

AVSZS[t_] := 0.02;

AV[t_]:= AVSZS[t]/Anull;

ca[t_]:= -AV[t];

cb[t_]:= -AV′ [t];

cc[t_]:= -I 2 κ^2;

d0[t_] := ComplexExpand[(-κ AV′[t] + I 2 κ^3 (1 + σ - t)) Exp[I χ]];

a0 := 0; b0 := 1; a1 := 1; b1 := σ; g0 := 0; g1 := 0;

dh := π/4; gh := 2; bh := π/2;

h := N[Log[π dh gh Ns/bh]/(gh Ns), NDEC];
```

---

In this part, we define function $\phi(x)$, its inverse function $\psi(t)$, and the sinc nodes $x_k$.

```
Psit[t_] := N[1/2 Tanh[(π/2) Sinh[t]] + 1/2, NDEC];

Psipt[t_] := N[π/4 Cosh[t]/((Cosh[π/2 Sinh[t]])^2), NDEC];

Phix[z_] := N[ArcSinh[2/π ArcTanh[2 z - 1]], NDEC];

Zk = N[Table[Psit[(i - Ns - 1) h], {i, 2 Ns + 1}], NDEC];
```

```
zn = Zk[[2 Ns + 1]];
zn = 1;
Tk = Zk/zn;
```

In this section, we define the exact solution of the problem. Later, we compare approximate solutions to the exact solution.

```
WC[t_] := ComplexExpand[(κ (1 - I) σ Cosh[κ (1 - I) (1 - t)] + Sinh[κ (1 - I) (1 - t)])/((1 - I) (Cosh[κ (1 - I)] + κ (1 - I) σ Sinh[κ (1 - I)]))];
U[t_] := Re[WC[t]] Cos[χ] - Im[WC[t]] Sin[χ];
V[t_] := Re[WC[t]] Sin[χ] + Im[WC[t]] Cos[χ];
UP[t_] := Re[WC′[t]] Cos[χ] - Im[WC′[t]] Sin[χ];
VP[t_] := Re[WC′[t]] Sin[χ] + Im[WC′[t]] Cos[χ];
UPP[t_] := Re[WC″[t]] Cos[χ] - Im[WC″[t]] Sin[χ];
VPP[t_] := Re[WC″[t]] Sin[χ] + Im[WC″[t]] Cos[χ];
U0 := (τ_w DE)/(ρ Anull);
vexact[t_] := U0 (U[t] + I V[t]);
```

Here the matrices of coefficents a(x), b(x), c(x) and d(x) are defined.

```
CA = N[Table[zn^2 ca[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CB = N[Table[zn cb[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CC = N[Table[cc[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CAD = N[DiagonalMatrix[CA], NDEC];
```

```
CBD = N[DiagonalMatrix[CB], NDEC];

CCD = N[DiagonalMatrix[CC], NDEC];

D0 = N[Table[d0[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
```

---

In this part, Hermit cardinal functions and their derivatives are defined.

```
H1[z_] = (-1 + z/zn)ˆ2 z;

H2[z_] = (-1 + z/zn)ˆ2 (1 + 2 z/zn);

H3[z_] = (z/zn)ˆ2 (3 - 2 z/zn);

H4[z_] = (-1 + z/zn) zˆ2/zn;

MH1 = Table[H1[Zk[[i]]], {i, 2 Ns + 1}];

MH2 = Table[H2[Zk[[i]]], {i, 2 Ns + 1}];

MH3 = Table[H3[Zk[[i]]], {i, 2 Ns + 1}];

MH4 = Table[H4[Zk[[i]]], {i, 2 Ns + 1}];

MH1p = Table[H1′[Zk[[i]]], {i, 2 Ns + 1}];

MH2p = Table[H2′[Zk[[i]]], {i, 2 Ns + 1}];

MH3p = Table[H3′[Zk[[i]]], {i, 2 Ns + 1}];

MH4p = Table[H4′[Zk[[i]]], {i, 2 Ns + 1}];

MH1pp = Table[H1″[Zk[[i]]], {i, 2 Ns + 1}];

MH2pp = Table[H2″[Zk[[i]]], {i, 2 Ns + 1}];

MH3pp = Table[H3″[Zk[[i]]], {i, 2 Ns + 1}];

MH4pp = Table[H4″[Zk[[i]]], {i, 2 Ns + 1}];
```

---

Here, $\delta_{j,k}^{(n)}$, for n = -1, 0, 1 and related coefficients to calculate the $M_{i,k}$ are defined.

```
PHIPINV = Table[Psipt[i h], {i, -Ns, Ns}];
DPPI = DiagonalMatrix[PHIPINV];
PHIP = Table[1/(Psipt[i h]), {i, -Ns, Ns}];
DP = DiagonalMatrix[PHIP];
E0 = Table[If[k == l, 1, 0], {k, -Ns, Ns}, {l, -Ns, Ns}];
I1 = Table[If[k == l, 0, (-1)^(k - l)/(k - l)], {k, -Ns, Ns}, {l, -Ns, Ns}];
IM1 = N[Table[If[k == l, 1/2, 1/2 + SinIntegral[π (k - l)]/π], {k, -Ns, Ns},
{l, -Ns, Ns}], NDEC];
B0 = h IM1.DPPI;
A0 = (1/h) DP.I1;
```

---

Here, the matrix $M_{i,k}$ is created.

```
LMH1 = CAD.MH1pp + CBD.MH1p + CCD.MH1;
LMH2 = CAD.MH2pp + CBD.MH2p + CCD.MH2;
LMH3 = CAD.MH3pp + CBD.MH3p + CCD.MH3;
LMH4 = CAD.MH4pp + CBD.MH4p + CCD.MH4;
VNM1 = CAD.A0 + CBD.E0 + CCD.B0;
VNM = ArrayFlatten[Table[{{Transpose[{LMH2}], Transpose[{LMH1}], VNM1,
Transpose[{LMH4}], Transpose[{LMH3}]}}]];
Z0 = Table[0, {i, 2 Ns + 1}];
```

```
B1 = N[Join[{a0, b0 zn}, Z0, {0, 0}], NDEC];
B2 = N[Join[{0, 0}, Z0, {b1 zn, a1}], NDEC];
B3 = N[Join[{0, 0}, Ln1, {0, 0}], NDEC];
B4 = N[Join[{0, 0}, Ln01, {0, 0}], NDEC];
Ln01 = h Table[(1/2 + N[SinIntegral[π (Ns + 1 - l)]/π, NDEC]) Psipt[l h], {l, -Ns, Ns}];
Ln1 = h Table[(1/2 + N[SinIntegral[π (-Ns - 1 - l)]/π, NDEC]) Psipt[l h], {l, -Ns, Ns}];
VNMe = Flatten[Table[{{B1, B2}, VNM, {B3, B4}}], 1];
D0e = Join[{g0, g1}, D0, {0, 0}];
```

In this part, we solve the main problem to approximate w(x). Afterwards, we derive u(x) and v(x) out of w(x).

```
vpne = N[LinearSolve[VNMe, D0e], NDEC];
vpns = Table[vpne[[i + 2]], {i, 2 Ns + 1}];
vpnsu = Table[vpns[[i]] + vpne[[1]] MH2p[[i]] + vpne[[2]] MH1p[[i]] + vpne[[2 Ns + 4]] MH4p[[i]] + vpne[[2 Ns + 5]] MH3p[[i]], {i, 2 Ns + 1}];
Rvpns = Table[Re[vpne[[i + 2]]], {i, 2 Ns + 1}];
Ivpns = Table[Im[vpne[[i + 2]]], {i, 2 Ns + 1}];
Rvpnsu = Table[Re[vpns[[i]]] + Re[vpne[[1]] MH2p[[i]]] + Re[vpne[[2]] MH1p[[i]]] + Re[vpne[[2 Ns + 4]] MH4p[[i]]] + Re[vpne[[2 Ns + 5]] MH3p[[i]]], {i, 2 Ns + 1}];
Ivpnsu = Table[Im[vpns[[i]]] + Im[vpne[[1]] MH2p[[i]]] + Im[vpne[[2]] MH1p[[i]]]
+ Im[vpne[[2 Ns + 4]] MH4p[[i]]] + Im[vpne[[2 Ns + 5]] MH3p[[i]]], {i, 2 Ns
+ 1}];
```

---

In the following sections, we try to plot the approximate solution against the exact solution.

```
coef1 = N[-κ Cos[χ], NDEC];
coef2 = N[-κ Sin[χ], NDEC];
Upexactplx = Table[{Zk[[i]], U0 UP[Zk[[i]]]}, {i, 2 Ns + 1}];
Rvpnsx = Table[{Zk[[i]], Rvpns[[i]]}, {i, 2 Ns + 1}];
Rvpnsux = Table[{Zk[[i]], U0 ( Rvpnsu[[i]] + coef1)}, {i, 2 Ns + 1}];
ListLinePlot[{Upexactplx, Rvpnsux}, PlotStyle → {Blue, Red}, PlotRange
→ {-0.5, 0.2}]

Vpexactplx = Table[{ Zk[[i]], U0 VP[Zk[[i]]]}, {i, 2 Ns + 1}];
Vvpnsx = Table[{Zk[[i]], Ivpns[[i]]}, {i, 2 Ns + 1}];
Vvpnsux = Table[{Zk[[i]], U0 (Ivpnsu[[i]] + coef2)}, {i, 2 Ns + 1}];
ListLinePlot[{Vpexactplx, Vvpnsux}, PlotStyle → {Blue, Red}]

vns = B0.vpns;
vnsu = Table[vns[[i]] + vpne[[1]] MH2[[i]] + vpne[[2]] MH1[[i]] + vpne[[2 Ns
+ 4]] MH4[[i]] + vpne[[2 Ns + 5]] MH3[[i]], {i, 2 Ns + 1}];
Rvns = Re[vns];
```

```
Rvnsu = Table[Re[vns[[i]]] + Re[vpne[[1]] MH2[[i]]] + Re[vpne[[2]] MH1[[i]]] + Re[vpne[[2 Ns + 4]] MH4[[i]]] + Re[vpne[[2 Ns + 5]] MH3[[i]]], {i, 2 Ns + 1}];
Ivns = Im[vns];
Ivnsu = Table[Im[vns[[i]]] + Im[vpne[[1]] MH2[[i]]] + Im[vpne[[2]] MH1[[i]]] + Im[vpne[[2 Ns + 4]] MH4[[i]]] + Im[vpne[[2 Ns + 5]] MH3[[i]]], {i, 2 Ns + 1}];

Uexactplx = Table[{Zk[[i]], U0 U[Tk[[i]]]}, {i, 2 Ns + 1}];
Rvnsx = Table[{Zk[[i]], Rvns[[i]]}, {i, 2 Ns + 1}];
Rvnsux = Table[{Zk[[i]], U0 (Rvnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Cos[χ], NDEC])}, {i, 2 Ns + 1}];
ListLinePlot[{Uexactplx, Rvnsux}, PlotStyle → {Blue, Red}, PlotRange → {-0.035, 0.002}]

Vexactplx = Table[{Zk[[i]], U0 V[Tk[[i]]]}, {i, 2 Ns + 1}];
Vvnsx = Table[{Zk[[i]], Ivns[[i]]}, {i, 2 Ns + 1}];
Vvnsux = Table[{Zk[[i]], U0 (Ivnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Sin[χ], NDEC])}, {i, 2 Ns + 1}];
ListLinePlot[{Vexactplx, Vvnsux}, PlotStyle → {Blue, Red}, PlotRange → {-0.01, 0.1}]

vppns = A0.vpns;
```

```
vppnsu = Table[vppns[[i]] + vpne[[1]] MH2pp[[i]] + vpne[[2]] MH1pp[[i]] +
vpne[[2 Ns + 4]] MH4pp[[i]] + vpne[[2 Ns + 5]] MH3pp[[i]], {i, 2 Ns + 1}];
Rvppns = Re[vppns];
Rppnsu = Table[Re[vppns[[i]]] + Re[vpne[[1]] MH2pp[[i]]] + Re[vpne[[2]] MH1pp[[i]]]
+ Re[vpne[[2 Ns + 4]] MH4pp[[i]]] + Re[vpne[[2 Ns + 5]] MH3pp[[i]]], {i, 2
Ns + 1}];
Ivppns = Im[vppns];
Ippnsu = Table[Im[vppns[[i]] + vpne[[1]] MH2pp[[i]] + vpne[[2]] MH1pp[[i]]
+ vpne[[2 Ns + 4]] MH4pp[[i]] + vpne[[2 Ns + 5]] MH3pp[[i]]], {i, 2 Ns + 1}];

Uppexactplx = Table[{Zk[[i]], U0 UPP[Tk[[i]]]}, {i, 2 Ns + 1}];
Rvppnsx = Table[{Zk[[i]], Rvppns[[i]]}, {i, 2 Ns + 1}];
Rvppnsux = Table[{Zk[[i]], U0 Rppnsu[[i]]}, {i, 2 Ns + 1}];
ListLinePlot[{Uppexactplx, Rvppnsux}, PlotStyle → {Blue, Red}]

Vppexactplx = Table[{Zk[[i]], U0 VPP[Tk[[i]]]}, {i, 2 Ns + 1}];
Ivppnsx = Table[{Zk[[i]], Ivppns[[i]]}, {i, 2 Ns + 1}];
Ivppnsux = Table[{Zk[[i]], U0 Ippnsu[[i]]}, {i, 2 Ns + 1}];
ListLinePlot[{Vppexactplx, Ivppnsux}, PlotStyle → {Blue, Red}, PlotRange
→ {-0.01, 1.7}]
```

Finally, we calculate the maximum error of our method.

```
RvppError = Table[{U0 Rppnsu[[i]] - U0 UPP[Tk[[i]]]}, {i, 2 Ns + 1}];
IvppError = Table[{U0 Ippnsu[[i]] - U0 VPP[Tk[[i]]]}, {i, 2 Ns + 1}];
RvpError = Table[{U0 (Rvpnsu[[i]] - N[κ Cos[χ], NDEC]) - U0 UP[Tk[[i]]]},
{i, 2 Ns + 1}];
IvpError = Table[{U0 (Ivpnsu[[i]] - N[κ Sin[χ], NDEC]) - U0 VP[Tk[[i]]]},
{i, 2 Ns + 1}];
RvError = Table[{U0 (Rvnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Cos[χ], NDEC]) -
U0 U[Zk[[i]]]}, {i, 2 Ns + 1}];
IvError = Table[{U0 (Ivnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Sin[χ], NDEC]) - U0
V[Zk[[i]]]}, {i, 2 Ns + 1}];
ErR1 = Max[Abs[RvError]];
ErI1 = Max[Abs[IvError]];
EdrR1 = Max[Abs[RvpError]];
EdrI1 = Max[Abs[IvpError]];
EddrR1 = Max[Abs[RvppError]];
EddrI1 = Max[Abs[IvppError]];
RvppErrorZk = Table[{Zk[[i]], U0 Rppnsu[[i]] - U0 UPP[Tk[[i]]]}, {i, 2 Ns +
1}];
IvppErrorZk = Table[{Zk[[i]], U0 Ippnsu[[i]] - U0 VPP[Tk[[i]]]}, {i, 2 Ns +
1}];
RvpErrorZk = Table[{Zk[[i]], U0 (Rvpnsu[[i]] - N[κ Cos[χ], NDEC]) - U0
UP[Tk[[i]]]}, {i, 2 Ns + 1}];
IvpErrorZk = Table[{Zk[[i]], U0 (Ivpnsu[[i]] - N[κ Sin[χ], NDEC]) - U0 VP[Tk[[i]]]},
{i, 2 Ns + 1}];
RvErrorZk = Table[{Zk[[i]], U0 (Rvnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Cos[χ],
NDEC]) - U0 U[Tk[[i]]]}, {i, 2 Ns + 1}];
IvErrorZk = Table[{Zk[[i]], U0 (Ivnsu[[i]] + N[κ (1 + σ - Zk[[i]]) Sin[χ],
NDEC]) - U0 V[Tk[[i]]]}, {i, 2 Ns + 1}];
ListLinePlot[{IvppErrorZk, IvpErrorZk, IvErrorZk, RvErrorZk, RvpErrorZk,
RvppErrorZk}, PlotRange → All]
```

## A.2 The Mathematica Source Code for the Coupled System

Here, the code written to solve the coupled system is shared. Since the steps are similar to those of the complex system, no explanation is added.

```
$MinPrecision = 16;
$MaxPrecision = 64;
NDEC := 32;
Ns := 8;
χ := π/4;
κ := 5;
DZ := 100;
DE := 20;
σ := 0.1;
ww := 9;
```

```
CD := 0.0012;

τw := 0.1414;

ρ := 1000;

Anull := 0.02;

AVS = 0.02;

AV[t_] := AVS/Anull;

ca1[t_] := -AV[t];

cb1[t_] := -AV'[t];

cc1[t_] := 2 κ^2;

d01[t_] := -2 κ^3 (1 + σ - t) Sin[χ] - κ Cos[χ] AV'[t];

a01 := 0; b01 := 1; a11 := 1; b11 := σ; g01 := 0; g11 := 0;

ca2[t_] := -AV[t];

cb2[t_] := -AV'[t];

cc2[t_] := -2 κ^2;

d02[t_] := 2 κ^3 (1 + σ - t) Cos[χ] - κ Sin[χ] AV'[t];

a02 := 0; b02 := 1; a12 := 1; b12 := σ; g02 := 0; g12 := 0;

dh := π/4; gh := 2; bh := π/2;

h := N[Log[π dh gh Ns/bh]/(gh Ns), NDEC];
```

```
Psit[t_] := N[1/2 Tanh[(π/2)*Sinh[t]] + 1/2, NDEC];

Psipt[t_] := N[π/4 Cosh[t]/((Cosh[π/2 Sinh[t]])^2), NDEC];

Phix[z_] := N[ArcSinh[2/π ArcTanh[2 z - 1]], NDEC];

Zk = N[Table[Psit[(i - Ns - 1) h], {i, 2 Ns + 1}], NDEC];
```

```
zn = Zk[[2 Ns + 1]];
zn = 1;
Tk = Zk/zn;
```

---

```
WC[t_] := ComplexExpand[(κ (1 - I) σ Cosh[κ (1 - I) (1 - t)] + Sinh[κ (1 - I) (1 - t)])/((1 - I) (Cosh[κ (1 - I)] + κ (1 - I) σ Sinh[κ (1 - I)]))];
U[t_] := ComplexExpand[Re[WC[t]] Cos[χ] - Im[WC[t]] Sin[χ]];
V[t_] := ComplexExpand[Re[WC[t]] Sin[χ] + Im[WC[t]] Cos[χ]];
U0 := (τ_w DE)/(ρ Anull);
exactsol[t_] := U0 (U[t] + I V[t]);
```

---

```
CA = N[Table[zn^2 ca1[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CB = N[Table[zn cb1[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CC1 = N[Table[cc1[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CC2 = N[Table[cc2[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
CAD = N[DiagonalMatrix[CA], NDEC];
CBD = N[DiagonalMatrix[CB], NDEC];
CCD1 = N[DiagonalMatrix[CC1], NDEC];
CCD2 = N[DiagonalMatrix[CC2], NDEC];
D01 = N[Table[d01[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
D02 = N[Table[d02[Zk[[i]]], {i, 2 Ns + 1}], NDEC];
```

---

```
H1[z_] = (-1 + z/zn)^2 z;
H2[z_] = (-1 + z/zn)^2 (1 + 2 z/zn);
H3[z_] = (z/zn)^2 (3 - 2 z/zn);
H4[z_] = (-1 + z/zn) z^2/zn;
MH1 = Table[H1[Zk[[i]]], {i, 2 Ns + 1}];
MH2 = Table[H2[Zk[[i]]], {i, 2 Ns + 1}];
MH3 = Table[H3[Zk[[i]]], {i, 2 Ns + 1}];
MH4 = Table[H4[Zk[[i]]], {i, 2 Ns + 1}];
MH1p = Table[H1′[Zk[[i]]], {i, 2 Ns + 1}];
MH2p = Table[H2′[Zk[[i]]], {i, 2 Ns + 1}];
MH3p = Table[H3′[Zk[[i]]], {i, 2 Ns + 1}];
MH4p = Table[H4′[Zk[[i]]], {i, 2 Ns + 1}];
MH1pp = Table[H1″[Zk[[i]]], {i, 2 Ns + 1}];
MH2pp = Table[H2″[Zk[[i]]], {i, 2 Ns + 1}];
MH3pp = Table[H3″[Zk[[i]]], {i, 2 Ns + 1}];
MH4pp = Table[H4″[Zk[[i]]], {i, 2 Ns + 1}];
```

---

```
PHIPINV = Table[Psipt[i h], {i, -Ns, Ns}];
DPPI = DiagonalMatrix[PHIPINV];
PHIP = Table[1/(Psipt[i h]), {i, -Ns, Ns}];
DP = DiagonalMatrix[PHIP];
E0 = Table[If[k == l, 1, 0], {k, -Ns, Ns}, {l, -Ns, Ns}];
IM1 = N[Table[If[k == l, 1/2, 1/2 + SinIntegral[π (k - l)]/π], {k, -Ns, Ns},
{l, -Ns, Ns}], NDEC];

I1 = Table[If[k == l, 0, (-1)^(k - l)/(k - l)], {k, -Ns, Ns}, {l, -Ns, Ns}];

B0 = h IM1 DPPI;

A0 = (1/h) DP I1;
```

---

```
L1MH1 = CAD.MH1pp + CBD.MH1p;

L1MH2 = CAD.MH2pp + CBD.MH2p;

L1MH3 = CAD.MH3pp + CBD.MH3p;

L1MH4 = CAD.MH4pp + CBD.MH4p;


L2MH1 = CAD.MH1pp + CBD.MH1p;

L2MH2 = CAD.MH2pp + CBD.MH2p;

L2MH3 = CAD.MH3pp + CBD.MH3p;

L2MH4 = CAD.MH4pp + CBD.MH4p;


LL1MH1 = CCD1.MH1;

LL1MH2 = CCD1.MH2;

LL1MH3 = CCD1.MH3;

LL1MH4 = CCD1.MH4;


LL2MH1 = CCD2.MH1;

LL2MH2 = CCD2.MH2;
```

```
LL2MH3 = CCD2.MH3;
LL2MH4 = CCD2.MH4;

VNM11 = CAD.A0 + CBD.E0;
VNM12 = CAD.A0 + CBD.E0;
VNM13 = CCD1.B0;
VNM14 = CCD2.B0;

Z0 = N[Table[0, {i, 2 Ns + 1}], NDEC];
Z00 = N[Table[0, {i, 2 Ns + 5}], NDEC];
Ln01 = h Table[(1/2 + N[SinIntegral[π (Ns + 1 - l)]/π, NDEC]) Psipt[l h],
{l, -Ns, Ns}];
Ln1 = h Table[(1/2 + N[SinIntegral[π (-Ns - 1 - l)]/π, NDEC]) Psipt[l h], {l,
-Ns, Ns}];
B1 = N[Join[{a01, b01 zn}, Z0, {0, 0}], NDEC];
B2 = N[Join[{0, 0}, Z0, {b11 zn, a11}], NDEC];
B3 = N[Join[{0, 0}, Ln1, {0, 0}], NDEC];
B4 = N[Join[{0, 0}, Ln01, {0, 0}], NDEC];

VNM1 = ArrayFlatten[Table[{Transpose[{L1MH2}], Transpose[{L1MH1}],
VNM11, Transpose[{L1MH4}], Transpose[{L1MH3}]}]];
VNM2 = ArrayFlatten[Table[{Transpose[{LL1MH2}], Transpose[{LL1MH1}],
VNM13, Transpose[{LL1MH4}], Transpose[{LL1MH3}]}]];
```

```
VNM3 = ArrayFlatten[Table[{Transpose[{LL2MH2}], Transpose[{LL2MH1}],
VNM14, Transpose[{LL2MH4}], Transpose[{LL2MH3}]}]];
VNM4 = ArrayFlatten[Table[{Transpose[{L2MH2}], Transpose[{L2MH1}],
VNM12, Transpose[{L2MH4}], Transpose[{L2MH3}]}]];

VNMe11 = Flatten[Table[{B1, B2}, VNM1, {B3, B4}], 1];
VNMe12 = Flatten[Table[{Z00, Z00}, VNM2, {Z00, Z00}], 1];
VNMe13 = Flatten[Table[{Z00, Z00}, VNM3, {Z00, Z00}], 1];
VNMe14 = Flatten[Table[{B1, B2}, VNM4, {B3, B4}], 1];

one = Join[VNMe11, VNMe12, 2];
two = Join[VNMe13, VNMe14, 2];
VNMe = Join[one, two, 1];
D0e1 = Join[{g01, g11}, D01, {0, 0}];
D0e2 = Join[{g02, g12}, D02, {0, 0}];
D0e = Join[D0e1, D0e2];
```

```
vpne = N[LinearSolve[VNMe, D0e], NDEC];
upns = Table[vpne[[i + 2]], {i, 2 Ns + 1}];
vpns = Table[vpne[[i + 2 Ns + 7]], {i, 2 Ns + 1}];
yupns = Table[upns[[i]] + vpne[[1]] MH2p[[i]] + vpne[[2]] MH1p[[i]] + vpne[[2
Ns + 4]] MH4p[[i]] + vpne[[2 Ns + 5]] MH3p[[i]], {i, 2 Ns + 1}];
yvpns = Table[vpns[[i]] + vpne[[2 Ns + 6]] MH2p[[i]] + vpne[[2 Ns + 7]]
MH1p[[i]] + vpne[[4 Ns + 9]] MH4p[[i]] + vpne[[4 Ns + 10]] MH3p[[i]], {i, 2
Ns + 1}];
```

---

```
yupexactplx = Table[{Zk[[i]], U0 U'[Tk[[i]]]}, {i, 2 Ns + 1}];
upnsx = Table[{Zk[[i]], upns[[i]]}, {i, 2 Ns + 1}];
yupnsx = Table[{Zk[[i]], U0 (yupns[[i]] - κ Cos[χ])}, {i, 2 Ns + 1}];
ListLinePlot[{yupexactplx, yupnsx}, PlotStyle → {Blue, Red}]

yvpexactplx = Table[{Zk[[i]], U0 V'[Tk[[i]]]}, {i, 2 Ns + 1}];
vpnsx = Table[{Zk[[i]], vpns[[i]]}, {i, 2 Ns + 1}];
yvpnsx = Table[{Zk[[i]], U0 (yvpns[[i]] - κ Sin[χ])}, {i, 2 Ns + 1}];
ListLinePlot[{yvpexactplx, yvpnsx}, PlotStyle → {Blue, Red}]

uns = B0.upns;
vns = B0.vpns;
yuns = Table[uns[[i]] + vpne[[1]] MH2[[i]] + vpne[[2]] MH1[[i]] + vpne[[2 Ns
+ 4]] MH4[[i]] + vpne[[2 Ns + 5]] MH3[[i]], {i, 2 Ns + 1}];
yvns = Table[vns[[i]] + vpne[[2 Ns + 6]] MH2[[i]] + vpne[[2 Ns + 7]] MH1[[i]]
+ vpne[[4 Ns + 9]] MH4[[i]] + vpne[[4 Ns + 10]] MH3[[i]], {i, 2 Ns + 1}];

yuexactplx = Table[{Tk[[i]], U0 U[Tk[[i]]]}, {i, 1, 2 Ns + 1}];
unsx = Table[{Zk[[i]], uns[[i]]}, {i, 2 Ns + 1}];
```

```
yunsx = Table[{Zk[[i]], U0 (yuns[[i]] + κ (1 + σ - Zk[[i]]) Cos[χ])}, {i, 1, 2
Ns + 1}];
ListLinePlot[{yuexactplx, yunsx}, PlotStyle → {Blue, Red}]

yvexactplx = Table[{Tk[[i]], U0 V[Tk[[i]]]}, {i, 1, 2 Ns + 1}];
vnsx = Table[{Zk[[i]], vns[[i]]}, {i, 2 Ns + 1}];
yvnsx = Table[{Zk[[i]], U0 (yvns[[i]] + κ (1 + σ - Zk[[i]]) Sin[χ])}, {i, 1, 2
Ns + 1}];
ListLinePlot[{yvexactplx, yvnsx}, PlotStyle → {Blue, Red}]

uppns = A0.upns;
vppns = A0.vpns;
yuppns = Table[uppns[[i]] + vpne[[1]] MH2pp[[i]] + vpne[[2]] MH1pp[[i]] +
vpne[[2 Ns + 4]] MH4pp[[i]] + vpne[[2 Ns + 5]] MH3pp[[i]], {i, 2 Ns + 1}];
yvppns = Table[vppns[[i]] + vpne[[2 Ns + 6]] MH2pp[[i]] + vpne[[2 Ns + 7]]
MH1pp[[i]] + vpne[[4 Ns + 9]] MH4pp[[i]] + vpne[[4 Ns + 10]] MH3pp[[i]],
{i, 2 Ns + 1}];

yuppexactplx = Table[{Zk[[i]], U0 U”[Tk[[i]]]}, {i, 1, 2 Ns + 1}];
uppnsx = Table[{Zk[[i]], uppns[[i]]}, {i, 2 Ns + 1}];
yuppnsx = Table[{Zk[[i]], U0 yuppns[[i]]}, {i, 1, 2 Ns + 1}];
ListLinePlot[{yuppexactplx, yuppnsx}, PlotStyle → {Blue, Red}]
```

```
yvppexactplx = Table[{Zk[[i]], U0 V”[Tk[[i]]]}, {i, 1, 2 Ns + 1}];
vppnsx = Table[{Zk[[i]], vppns[[i]]}, {i, 2 Ns + 1}];
yvppnsx = Table[{Zk[[i]], U0 yvppns[[i]]}, {i, 1, 2 Ns + 1}];
ListLinePlot[{yvppexactplx, yvppnsx}, PlotStyle → {Blue, Red}]
```

---

```
UvppError = Table[{U0 yuppns[[i]] - U0 U”[Tk[[i]]]}, {i, 2 Ns + 1}];
VvppError = Table[{U0 yvppns[[i]] - U0 V”[Tk[[i]]]}, {i, 2 Ns + 1}];
UvpError = Table[{U0 (yupns[[i]] - κ Cos[χ]) - U0 U’[Tk[[i]]]}, {i, 2 Ns +
1}];
VvpError = Table[{U0 (yvpns[[i]] - κ Sin[χ]) - U0 V’[Tk[[i]]]}, {i, 2 Ns + 1}];
UvError = Table[{U0 (yuns[[i]] + κ (1 + σ - Zk[[i]]) Cos[χ]) - U0 U[Tk[[i]]]},
{i, 2 Ns + 1}];
VvError = Table[{U0 (yvns[[i]] + κ (1 + σ - Zk[[i]]) Sin[χ]) - U0 V[Tk[[i]]]},
{i, 2 Ns + 1}];
ErU1 = Max[Abs[UvError]] ErV1 = Max[Abs[VvError]] EdrU1 = Max[Abs[UvpError]]
EdrV1 = Max[Abs[VvpError]] EddrU1 = Max[Abs[UvppError]] EddrV1 =
Max[Abs[VvppError]] UvppErrorZk = Table[{Zk[[i]], U0 yuppns[[i]] - U0
U”[Tk[[i]]]}, {i, 2 Ns + 1}];
VvppErrorZk = Table[{Zk[[i]], U0 yvppns[[i]] - U0 V”[Tk[[i]]]}, {i, 2 Ns +
1}];
UvpErrorZk = Table[{Zk[[i]], U0 (yupns[[i]] - κ Cos[χ]) - U0 U’[Tk[[i]]]}, {i,
2 Ns + 1}];
VvpErrorZk = Table[{Zk[[i]], U0 (yvpns[[i]] - κ Sin[χ]) - U0 V’[Tk[[i]]]}, {i,
2 Ns + 1}];
UvErrorZk = Table[{Zk[[i]], U0 (yuns[[i]] + κ (1 + σ - Zk[[i]]) Cos[χ]) - U0 U[Tk[[i]]]}, {i, 2 Ns + 1}];
VvErrorZk = Table[{Zk[[i]], U0 (yvns[[i]] + κ (1 + σ - Zk[[i]]) Sin[χ]) - U0 V[Tk[[i]]]}, {i, 2 Ns + 1}];
ListLinePlot[{UvppErrorZk, UvpErrorZk, UvErrorZk, VvErrorZk, VvpErrorZk, VvppErrorZk}, PlotRange → All]
```

# Bibliography

[1] K. Abdella, X. Yu, and I. Kucuk. Application of the sinc method to a dynamic elasto-plastic problem. *Journal of Computational and Applied Mathematics*, 223(2):626–645, 2009.

[2] Kenzu Abdella. Numerical solution of two-point boundary value problems using sinc interpolation. In *Proceedings of the American Conference on Applied Mathematics (American-Math'12): Applied Mathematics in Electrical and Computer Engineering*, pages 157–162, 2012.

[3] G Adomian. A review of the decomposition method and some recent results for nonlinear equations. *Mathematical and Computer Modelling*, 13(7):17–43, 1990.

[4] Kamel Al-Khaled. Numerical approximations for population growth models. *Applied mathematics and computation*, 160(3):865–873, 2005.

[5] Javed Ali. One dimensional differential transform method for some higher order boundary value problems in finite domain. *Int. J. Contemp. Math. Sciences*, 7(6):263–272, 2012.

[6] Basem S Attili and Muhammed I Syam. Efficient shooting method for solving two point boundary value problems. *Chaos, Solitons & Fractals*, 35(5):895–903, 2008.

[7] Owe Axelsson and Vincent Allan Barker. *Finite element solution of boundary value problems: theory and computation*, volume 35. Society for Industrial and Applied Mathematics, 1987.

[8] B. Bialecki. Sinc-collection methods for two-point boundary value problems. *IMA Journal of Numerical Analysis*, 11(3):357–375, 1991.

[9] RL Burden and JD Faires. Numerical analysis 7th ed., brooks/cole, thomson learning, 2001.

[10] CL Chen and YC Liu. Solution of two-point boundary-value problems using the differential transformation method. *Journal of Optimization Theory and Applications*, 99(1):23–35, 1998.

[11] Liz Creel. *Ripple effects: population and coastal regions*. Population Reference Bureau Washington, 2003.

[12] ZJ Csendes. A novel finite element method for two-point boundary value problems. *Mathematics and Computers in Simulation*, 20(3):197–203, 1978.

[13] Alan M Davies. Solution of the 3d linear hydrodynamic equations using an enhanced eigenfunction approach. *International journal for numerical methods in fluids*, 13(2):235–250, 1991.

[14] AM Davies. The numerical solution of the three-dimensional hydrodynamic equations, using a b-spline representation of the vertical current profile. *Elsevier Oceanography Series*, 19:1–25, 1977.

[15] AM Davies and A Owen. Three dimensional numerical sea model using the galerkin method with a polynomial basis set. *Applied mathematical modelling*, 3(6):421–428, 1979.

[16] Mehdi Dehghan and Abbas Saadatmandi. The numerical solution of a nonlinear system of second-order boundary value problems using the sinc-collocation method. *Mathematical and Computer Modelling*, 46(11): 1434–1441, 2007.

[17] V Walfrid Ekman. On the influence of the earth\'s rotation on ocean currents. *Ark. Mat. Astron. Fys.*, 2:1–53, 1905.

[18] M El-Gamel and AI Zayed. Sinc-galerkin method for solving nonlinear boundary-value problems. *Computers & Mathematics with Applications*, 48(9):1285–1298, 2004.

[19] Mohamed El-Gamel, SH Behiry, and H Hashish. Numerical method for the solution of special nonlinear fourth-order boundary value problems. *Applied Mathematics and Computation*, 145(2):717–734, 2003.

[20] MA El-Gebeily and BS Attili. An iterative shooting method for a certain class of singular two-point boundary value problems. *Computers & Mathematics with Applications*, 45(1):69–76, 2003.

[21] JD Faires and RL Burden. Numerical methods pacific grove, ca: Brooks/cole pub, 1998.

[22] Paul C Gierke. Discrete approximations of differential operators by sinc methods. 1999.

[23] Sung N Ha. A nonlinear shooting method for two-point boundary value problems. *Computers & Mathematics with Applications*, 42(10):1411–1420, 2001.

[24] Qun Han, Jiping Ning, Zhiqiang Chen, Lianju Shang, and Guofang Fan. An efficient shooting method for fibre raman amplifier design. *Journal of Optics A: Pure and Applied Optics*, 7(8):386, 2005.

[25] Ji-Huan He. Variational approach to the sixth-order boundary value problems. *Applied Mathematics and Computation*, 143(2):537–538, 2003.

[26] NS Heaps. Three-dimensional model for tides and surges with vertical eddy viscosity prescribed in two layersi. mathematical formulation. *Geophysical Journal of the Royal Astronomical Society*, 64(1):291–302, 1981.

[27] NS Heaps et al. On the numerical solution of the three-dimensional hydrodynamical equations for tides and storm surges. *Mémoires de la Société Royale des Sciences de Liège. Sixième Série*, 1972.

[28] E Hesameddini and E Asadolahifard. The sinc-collocation method for solving the telegraph equation. *Journal of Computer Engineering and Informatics*.

[29] Nguyen Trung Hieu. Remarks on the shooting method for nonlinear two-point boundary-value problems.

[30] Raymond W Holsapple. A modified simple shooting method for solving two-point boundary value problems. 2012.

[31] Jochen Kampf and Jochen Kämpf. *Ocean Modelling for Beginners: Using Open-source Software*. Springer, 2009.

[32] Fritz Keinert. Uniform approximation to x $\beta$ by sinc functions. *Journal of approximation theory*, 66(1):44–52, 1991.

[33] Herbert B Keller. Numerical methods for two-point boundary value problems. 1968. *Ginn-Blaisdel, Walthman, Mass.*

[34] Herbert B Keller. *Numerical solution of two point boundary value problems*. Siam, 1976.

[35] Sanoe Koonprasert. *THE SINC-GALERKIN METHOD FOR PROBLEMS*. PhD thesis, MONTANA STATE UNIVERSITY Bozeman, 2003.

[36] Sanoe Koonprasert and Kenneth L Bowers. Block matrix sinc-galerkin solution of the wind-driven current problem. *Applied mathematics and computation*, 155(3):607–635, 2004.

[37] Marek A Kowalski, Krzystof A Sikorski, and Frank Stenger. *Selected topics in approximation and computation*, volume 12. OUP USA, 1995.

[38] WF Langford. A shooting algorithm for the best least squares solution of two-point boundary value problems. *SIAM Journal on Numerical Analysis*, 14(3):527–542, 1977.

[39] RW Lardner and Y Song. A hybrid spectral method for the three-dimensional numerical modelling of nonlinear flows in shallow seas. *Journal of Computational Physics*, 100(2):322–334, 1992.

[40] Hua Li. Solving a nonlinear two-point boundary value problem. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 1038–1042. IEEE, 1992.

[41] Songxin Liang and David J Jeffrey. An analytical approach for solving nonlinear boundary value problems in finite domains. *Numerical Algorithms*, 56(1):93–106, 2011.

[42] Shijun Liao. *Beyond perturbation: introduction to the homotopy analysis method*, volume 2. Chapman and Hall/CRC, 2003.

[43] Andreas Lippke. Analytical solutions and sinc function approximations in thermal conduction with nonlinear heat generation. *Journal of Heat Transfer (Transcations of the ASME (American Society of Mechanical Engineers), Series C);(United States)*, 113(1), 1991.

[44] John Lund and Kenneth L Bowers. *Sinc methods for quadrature and differential equations*. Number 32. Society for Industrial Mathematics, 1987.

[45] John Lund and Curtis R Vogel. A fully-galerkin method for the numerical solution of an inverse problem in a parabolic partial differential equation. *Inverse Problems*, 6(2):205, 1999.

[46] Vasile Marinca, Nicolae Herişanu, and Iacob Nemeş. Optimal homotopy asymptotic method with application to thin film flow. *Central European Journal of Physics*, 6(3):648–653, 2008.

[47] Adel Mohsen and Mohamed El-Gamel. On the numerical solution of linear and nonlinear volterra integral and integro-differential equations. *Applied Mathematics and Computation*, 217(7):3330–3337, 2010.

[48] David D Morrison, James D Riley, and John F Zancanaro. Multiple shooting method for two-point boundary value problems. *Communications of the ACM*, 5(12):613–614, 1962.

[49] S. Narasimhan, J. Majdalani, and F. Stenger. A first step in applying the sinc collocation method to the nonlinear navier-stokes equations. *Numerical Heat Transfer: Part B: Fundamentals*, 41(5):447–462, 2002.

[50] S Natesan and N Ramanujam. shooting methodfor singularly perturbed one-dimensional reaction-diffusion neumann problems. *International journal of computer mathematics*, 72(3):383–393, 1999.

[51] Ali H Nayfeh. *Perturbation methods*. Wiley-VCH, 2008.

[52] M Aslam Noor and EA Al-Said. Finite-difference method for a system of third-order boundary-value problems. *Journal of optimization theory and applications*, 112(3):627–637, 2002.

[53] Mike R Osborne. On shooting methods for boundary value problems(shooting method difficulties in boundary value problem numerical

solutions using computer programs to integrate differential equations). *Journal of mathematical analysis and applications*, 27:417–433, 1969.

[54] K Parand and A Pirkhedri. Sinc-collocation method for solving astrophysics equations. *New Astronomy*, 15(6):533–537, 2010.

[55] Michael B Porter and Edward L Reiss. A note on the relationship between finite-difference and shooting methods for ode eigenvalue problems. *SIAM journal on numerical analysis*, 23(5):1034–1039, 1986.

[56] WH Press, BP Flannery, SA Teukolsky, and WT Vetterling. Numerical recipes (fortran version) cambridge university press. *Cambridge, MA*, 1989.

[57] N Ramanujam and S Natesan. shooting method'for singular perturbation problems arising in chemical reactor theory. *International journal of computer mathematics*, 70(2):251–262, 1998.

[58] J Rashidinia and Mehdi Ghasemi. B-spline collocation for solution of two-point boundary value problems. *Journal of computational and applied mathematics*, 235(8):2325–2342, 2011.

[59] J Rashidinia and M Zarebnia. The numerical solution of integro-differential equation by means of the sinc method. *Applied mathematics and computation*, 188(2):1124–1130, 2007.

[60] K Ruotsalainen and W Wendland. On the boundary element method for some nonlinear boundary value problems. *Numerische Mathematik*, 53(3):299–314, 1988.

[61] A Saadatmandi and M Razzaghi. The numerical solution of third-order boundary value problems using sinc-collocation method. *Communications in numerical methods in engineering*, 23(7):681–689, 2007.

[62] A Saadatmandi, M Razzaghi, and M Dehghan. Sinc-collocation methods for the solution of hallen's integral equation. *Journal of electromagnetic waves and applications*, 19(2):245–256, 2005.

[63] Abbas Saadatmandi. Numerical study of second painlevé equation. *Communications in Numerical Analysis*, 2012, 2012.

[64] Lawrence F Shampine. Singular boundary value problems for odes. *Applied mathematics and computation*, 138(1):99–112, 2003.

[65] Ralph C Smith and Kenneth L Bowers. Sinc-galerkin estimation of diffusivity in parabolic problems. *Inverse problems*, 9(1):113, 1999.

[66] F. Stenger. Summary of sinc numerical methods. *Journal of Computational and Applied Mathematics*, 121(1):379–420, 2000.

[67] Frank Stenger. *Numerical methods based on Sinc and analytic functions.* Springer-Verlag New York, 1993.

[68] Frank Stenger and Michael J O'Reilly. Computing solutions to medical problems via sinc convolution. *Automatic Control, IEEE Transactions on*, 43(6):843–848, 1998.

[69] Josef Stoer, Roland Bulirsch, R Bartels, Walter Gautschi, and Christoph Witzgall. *Introduction to numerical analysis*, volume 2. Springer New York, 1993.

[70] John Macnaghten Whittaker. *Interpolatory function theory.* Cambridge University Press Cambridge, 1935.

[71] DF Winter, Kenneth L Bowers, and John Lund. Wind-driven currents in a sea with a variable eddy viscosity calculated via a sinc–galerkin technique. *International journal for numerical methods in fluids*, 33(7): 1041–1073, 2000.