A NEW NUMERICAL APPROACH TO SOLVE THE 1D VISCOUS PLASTIC SEA ICE MOMENTUM EQUATION

Fahim Alam

BSc, Shahjalal University of Science & Technology, Bangladesh, 2017

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in the Department of Mathematics and
Statistics

©Fahim Alam, 2023

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Supervisory Committee

Dr. Boualem Khouider, Supervisor

Department of Mathematics and Statistics

University of Victoria

Dr. David Goluskin, Departmental Member

Department of Mathematics and Statistics

University of Victoria

Abstract

While there has been a colossal effort in the ongoing decades, the ability to simulate ocean ice has fallen behind various parts of the climate system and most Earth System Models are unable to capture the observed adversities of Arctic sea ice, which is, as it were, attributed to our frailty to determine sea ice dynamics. Viscous Plastic rheology is the most by and large recognized model for sea ice dynamics and it is expressed as a set of partial differential equations that are hard to tackle numerically. Using the 1D sea ice momentum equation as a prototype, we use the method of lines based on Euler's backward method. This results in a nonlinear PDE in space only. At that point, we apply the Damped Newton's method which has been introduced in Looper and Rapetti et al. [5] and used and generalized to 2D in Saumier et al. [2] to solve the Monge-Ampere equation. However, in our case, we need to solve 2nd order linear equation with discontinuous coefficients during Newton iteration. To overcome this difficulty, we use the Finite element method to solve the linear PDE at each Newton iteration. In this paper, we show that with the adequate smoothing and re-scaling of the linear equation, convergence can be guaranteed and the numerical solution indeed converges efficiently to the continuum solution unlike other numerical approaches that typically solve an alternate set of equations and avoid the difficulty of the Newton method for a large nonlinear algebraic system. The finite element solver failed to converge when the original setting of the smoothed SIME with a smoothing constant $K = 2.8 \times 10^8$ was used. A much smaller constant of K=100 was necessary. The large smoothing constant K leads to an ill conditioned mass matrix.

Contents

List of Tables

List of Figures

Acknowledgment:

I want to take a moment to express my deep appreciation for the incredible support and guidance I received from my supervisor, Dr. Boualem Khouider, during the entire research process for my thesis. Without his unwavering patience and support, I honestly don't think I would have been able to complete this challenging endeavour. Dr. Khouider has been more than just a mentor to me; he has been a constant source of encouragement and understanding, even during the toughest times in my life. He not only provided me with the financial assistance I needed but also offered emotional support when I needed it most. I truly consider him to be like a second father to me.

From the very beginning, Dr. Khouider has been there for me, helping me choose a research topic that would make a meaningful contribution. He went above and beyond by connecting me with the necessary resources and tools to acquire the results I needed. Whenever I had questions or concerns, he was always just an email away, responding almost immediately. I'll never forget the time he took out of his vacation to Skype with me and address my concerns. His dedication and mentorship have been truly exceptional, and I am forever grateful for his guidance throughout this journey.

I would also like to extend my heartfelt gratitude to Dr. David Goluskin for being a part of my thesis supervisory committee. His expertise and valuable input have shaped my research in significant ways, and I am grateful for his contributions.

Furthermore, I would like to express my sincere appreciation to Dr. Alexander Bihlo for taking on the role of my external examiner. His fresh perspective and insightful feedback have added tremendous value to my work.

Lastly, I want to acknowledge the immense support and love from my parents, MD Ashraful Alam and Shirin Sultana, as well as my wonderful wife, Shazia Zaman, and my daughter,

Arshiya Zaman Alam. Their unwavering belief in me, understanding of the challenges I faced, and constant support have been my driving force throughout this degree. I owe a great deal of my success to their love and encouragement.

To all these remarkable individuals who have played pivotal roles in my academic journey, I want to express my deepest gratitude. Your contributions and support have made a lasting impact on me, and I am truly blessed to have had each one of you by my side.

Chapter 1

Introduction:

While there has been a colossal effort in the ongoing decades, the ability to simulate ocean ice has fallen behind various parts of the climate system and most Earth System Models are unable to capture the observed adversities of Arctic sea ice, which is, as it were, attributed to our frailty to determine sea ice dynamics. The most challenging part while solving sea ice momentum comes from the rheology term which shows the connection between stress and strain. Although there are so many approaches have been considered to deal with these terms, Viscous Plastic rheology is the most recognized model for sea ice dynamics so far. Due to the possibility of rapid changes in viscosity, this momentum equation is a stiff equation under this approach. Hence using an implicit approach will be an ideal choice to deal with the problem. Hibler III [9] solved the equation using that method. He used Picard iteration with Successive Over- Relaxation (SOR). But it takes many iteration steps to converge. Then, Lemieux et al.[10] coined a new version of Newton's method(Jacobian Free Newton's method) which can solve the nonlinear problem with better accuracy. Following that Seinen and Khouider [1] improved the method by using a different technique. They have used a second-order Crank-Nicolson scheme in stead of using the backward Euler scheme and also an improved Jacobian approximation. Mehlmann and Richter [12] used finite element multigrid-framework to solve the sea ice momentum equation. For our research, we choose the 1D sea ice momentum equation as a prototype, using the method of lines based on Euler's backward scheme. This results in a nonlinear PDE in space only. At that point, we apply the Damped Newton's method which has been introduced in Looper

and Rapetti et al. [5] and used and generalized to 2D in Saumier et al. [2] to solve the Monge-Ampere equation. However, in our case, we need to solve 2nd-order linear equation with discontinuous coefficients during Newton iteration. To overcome this difficulty, we use the Finite element method [11] to solve the linear PDE at each Newton iteration. In this paper, we will show that with the adequate smoothing and re-scaling of the linear equation, convergence can be guaranteed and the numerical solution indeed converges efficiently to the continuum solution, unlike other numerical approaches that typically solve an alternate set of equations and avoid the difficulty of the Newton method for a large nonlinear algebraic system.

1.1 Outline:

This thesis is structured as follows. In Chapter 2, we introduce the Viscous-Plastic sea ice momentum equation. In Chapter 3, we present our numerical design to tackle the problem. In Chapter 4, we discuss how to deal with linear differential equations with discontinuous co-efficient and how to solve them using the Finite element method. In Chapter 5, we show an analysis of our solver in detail. In chapter 6, we establish that the solver is 2nd order convergence in space and 1st order in temporary resolution using toy problems. Finally, in Chapter 6, we perform extensive numerical tests on SIME using a synthetic solution, validating the solver and confirming fully second-order convergence. We conclude with closing remarks and highlight the key takeaways from this study in Chapter 7.

Chapter 2

Viscous Plastic Sea Ice Momentum Equation

The momentum equation is considered in general as a two-dimensional problem (neglecting motions and forces out of the sea surface plane)and it can be written as

$$\rho h \frac{D\mathbf{u}}{Dt} = -\rho h f \boldsymbol{k} \times \mathbf{u} + \tau_a - \tau_w + \nabla \cdot \sigma - \rho h g \nabla H_d, \tag{2.1}$$

where,

$\mathbf{u} =$ horizontal velocity vector of sea ice $= u\boldsymbol{i} + v\boldsymbol{j}$,

$h =$ the sea-ice thickness(ice volume per unit area),

$f =$ Coriolis parameter,

$\boldsymbol{k} =$ unit vector perpendicular to the horizontal plane

$\tau_a =$ Wind stress applied on the ice surface

$\tau_w =$ Water stress applied on the ice underside

$\sigma =$ Internal ice-stress tensor

$g =$ Acceleration due to gravity

$\rho =$ Sea Ice Density

$H_d =$ Sea surface height

In (2.1), $\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}$ is the material derivative with respect to the ice motion. Because the pack ice velocities are relatively small, the contribution of the non-linear advection typically neglected.

The sea surface tilt is expressed in terms of the geostrophic ocean current $\mathbf{u}_w^g$, i.e.

$$-f\boldsymbol{k} \times \mathbf{u}_w^g = g\nabla H_d. \tag{2.2}$$

The air and water drag terms are expressed with constant turning angles,

$$\tau_a = \rho_a C_{da} |\mathbf{u}_a^g| (\mathbf{u}_a^g \cos\theta_a + (\boldsymbol{k} \times \mathbf{u}_a^g) \sin\theta_a), \tag{2.3}$$

$$\tau_w = C_w((\mathbf{u} - \mathbf{u}_w^g)\cos\theta_w + (\boldsymbol{k} \times (\mathbf{u} - \mathbf{u}_w^g))\sin\theta_w), \tag{2.4}$$

where

$$C_w = \rho_w C_{dw} |\mathbf{u} - \mathbf{u}_w^g|. \tag{2.5}$$

Here, ρ_a and ρ_w is the wind and water densities respectively and C_{da} and C_{dw} are air and water drag coefficients. And u_w^g stands for geostrophic ocean velocities. We observe that sea velocity is neglected in the formulation of wind stress.

Now to model the ice interaction term(or the rheology term), we can relate internal stress and strain rates by,

$$\sigma_{ij} = 2\eta\dot{\epsilon}_{ij} + \big[\zeta - \eta\big]\dot{\epsilon}_{kk}\delta_{ij} - \frac{P\delta_{ij}}{2}, \quad i,j = 1,2. \tag{2.6}$$

where σ_{ij} is the 2D internal stress tensor, ζ and η are the nonlinear bulk and shear viscosities respectively and $\dot{\epsilon}$ is the strain rates and $\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$

The strain rates are defined as,

$$\dot{\epsilon}_{11} = \frac{\partial u}{\partial x}, \dot{\epsilon}_{22} = \frac{\partial u}{\partial y}, \dot{\epsilon}_{12} = \dot{\epsilon}_{21} = \frac{1}{2}\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right), \dot{\epsilon}_{kk} = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}. \tag{2.7}$$

The hydrostatic ice pressure or P is given by,

$$P = P^* h \cdot \exp[-C(1-A)], \tag{2.8}$$

where $P^* = 27.5 \times 10^3 \mathrm{N}m^{-2}$ is the ice strength parameter and $C = 20$, the ice concentration parameter which characterizes the dependence of compression strength by ice on the area

fraction, A. Lastly, the nonlinear bulk and shear viscosity η and ζ are functions of ice strain rate invariant and ice strength.

In this study, we consider the 1D version of (2.1) which is,

$$\rho h \frac{\partial u}{\partial t} = \tau_a - \tau_w + \frac{\partial \sigma}{\partial x}. \tag{2.9}$$

We redefine $\sigma = \sigma_{11}$. Also, we get the following changes as well,

$$\tau_a = \rho_a C_{da} |u_a| u_a, \tag{2.10}$$

$$\tau_w = \rho_w C_{dw} |u - u_w| (u - u_w), \tag{2.11}$$

$$\frac{\partial \sigma_{11}}{\partial x} = \frac{\partial}{\partial x}\left[(\zeta + \eta)\frac{\partial u}{\partial x}\right] - \frac{1}{2}\frac{\partial P}{\partial x}, \tag{2.12}$$

$$\text{where } \zeta = \frac{P}{2\Delta} \tag{2.13}$$

$$\text{and } \eta = \zeta e^{-2},\ e \text{ is the eccentricity of the sea ice yield curve=2.} \tag{2.14}$$

Lastly, Δ can be written as,

$$\Delta = \left[\left(1 + e^{-2}\right)\left(\frac{\partial u}{\partial x}\right)^2\right]^{\frac{1}{2}} = (1 + e^{-2})^{\frac{1}{2}} \left|\frac{\partial u}{\partial x}\right|. \tag{2.15}$$

In order to make sure that Δ never becomes 0, we follow [1] and redefine bulk viscosity as,

$$\zeta \equiv \zeta_{max} \tanh\left(\frac{P}{2\zeta_{max}\Delta}\right) = kP \tanh\left(\frac{1}{2k\Delta}\right), k = 10^2,\ \zeta_{max} = kP. \tag{2.16}$$

And lastly continuity equations for 1D case are(in the absence of thermodynamic source term),

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0, \tag{2.17}$$

$$\frac{\partial A}{\partial t} + \frac{\partial}{\partial x}(Au) = 0, \tag{2.18}$$

which describes the thickness of the sea ice (h) and the area fraction of the thick ice (A) are advected by u.

Chapter 3

Numerical Design

To solve (2.1), We imply a Backward Euler difference on temporal resolution and then instead of discretizing the equation, we linearize it term by term. We observe that viscosities have absolute terms and we deal with this by considering every possible case for the term. Lastly, we apply the Damped Newton's method which already has been introduced in Looper and Rapetti [5] and also in Saumier [2].

3.1 Numerical Approach

Reriting (2.9), we get,

$$-\rho h\frac{\partial u}{\partial t}+\tau_a-\tau_w+\frac{\partial}{\partial x}\left[(1+e^{-2})\zeta\frac{\partial u}{\partial x}\right]-\frac{1}{2}\frac{\partial P}{\partial x}=0,\ \text{where, } \eta=\zeta e^{-2}. \tag{3.1}$$

Using a backward Euler difference on temporal resolution δt, we solve (3.1) at times $\delta t, 2\delta t, \cdots, k\delta t, \cdots, T$, identifying variables at time level k via superscripts, the equation (3.1), becomes,

$$-\rho h\frac{u^k-u^{k-1}}{\delta t}+\tau_a^k-\tau_w^k+\frac{\partial}{\partial x}\left[(1+e^{-2})\zeta\frac{\partial u}{\partial x}\right]^k-\frac{1}{2}\left[\frac{\partial P}{\partial x}\right]^k=0. \tag{3.2}$$

Expanding the water drag and rearranging the terms,(3.2) can be written as

$$-\rho h\frac{u^k}{\delta t}-\rho_w C_{dw}|u^k-u_w^k|(u^k-u_w^k)+\frac{\partial}{\partial x}\left[(1+e^{-2})\zeta\frac{\partial u}{\partial x}\right]^k=r_*^k. \tag{3.3}$$

where $r_*^k = \left[\frac{\partial P}{\partial x}\right]^k + \tau_a^k - \rho h \frac{u^{k-1}}{\delta t}$ and $\frac{\partial P}{\partial x} = P^* \cdot \exp[-C(1-A)]\left(\frac{\partial h}{\partial x} - Ch\frac{\partial A}{\partial x}\right)$.
Dropping superscript k,(4.3) becomes,

$$\frac{-\rho h}{\delta t}u - \rho_w C_{dw}|u-u_w|(u-u_w) + (1+e^{-2})\frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial x} + (1+e^{-2})\zeta\frac{\partial^2 u}{\partial x^2} = r_*. \tag{3.4}$$

We can rewrite (3.4) as

$$M(u) = f(x). \tag{3.5}$$

where, $M(u) = \frac{-\rho h}{\delta t}u - \rho_w C_{dw}|u-u_w|(u-u_w) + (1+e^{-2})\frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial x} + (1+e^{-2})\zeta\frac{\partial^2 u}{\partial x^2}$ and $f(x) = r_*$.
To solve (3.4), we will apply Damped Newton's method following [2].

3.1.1 Damped Newton's method

$$\begin{cases} \text{With the initial value of } u \; i.e \; u^0, \text{ where } u^0 = u^{k-1} \text{ and initial guess } u_0 \text{ in (3.2),} \\ \text{Solve for } v_n, \mathcal{L}(u_n)\cdot v_n = \frac{1}{\tau}(f^k - f_n), \text{ where } f^k = r_*^k, f_n = M(u_n). \\ \text{Update the solution using, } u_{n+1} = u_n + v_n. \end{cases}$$

where $\mathcal{L}(u_n)$ is linear differential equation of order n in direction of v_n and $\frac{1}{\tau}(\tau > 1)$ is a step-size parameter.
Now to use this method, we need to linearize (3.4).

3.2 Linearizing the SIME equation:

Using Backward Euler on temporal resolution, we get,

$$u^k + \mathcal{N}\left[\rho_w C_{dw}|u^k - u_w|(u^k - u_w) - E^2\zeta_x^k u_x^k. - E^2\zeta^k u_{xx}^k\right] = u^{k-1},$$

assuming, $\mathcal{N} = \frac{\triangle t}{\rho h}; E = (1+e^{-2})^{\frac{1}{2}}$.

3.2.1 Setting up the equation:

We consider,

$$\hat{a}(u) = u + \underbrace{\mathcal{N}\rho_w C_{dw}}_{\mathcal{N}_w}|u - u_w|(u - u_w),$$
$$\hat{b}(u) = \mathcal{N}E^2\zeta_x u_x,$$
$$\hat{c}(u) = \mathcal{N}E^2\zeta u_{xx}.$$

We will linearize $\hat{a}, \hat{b}$ and $\hat{c}$. Let, $\epsilon_1 = \text{sign}(u - u_w)$ and $\epsilon > 0$ is small.
We have,

$$\begin{aligned}
\hat{a}(u + \epsilon v) &= (u + \epsilon v) + \mathcal{N}_w\epsilon_1(u - u_w + \epsilon v)^2, \\
&= u + \epsilon v + \mathcal{N}_w\epsilon_1\Big[(u - u_w)^2 + 2\epsilon v(u - u_w) + \epsilon^2 v^2\Big], \\
&= \underbrace{u + \mathcal{N}_w|u - u_w|(u - u_w)}_{a(u)} + \Big[2\mathcal{N}_w|u - u_w| + 1\Big]\epsilon v + \mathsf{o}(\epsilon), \\
&= \hat{a}(u) + \underbrace{\Big[2\mathcal{N}_w|u - u_w| + 1\Big]}_{a_1}\epsilon v + \mathsf{o}(\epsilon), \\
&= \hat{a}(u) + a_1\epsilon v + \mathsf{o}(\epsilon).
\end{aligned}$$

Now, when $u_x \neq 0$,

$$b(u + \epsilon v) = \mathcal{N}E^2\zeta_x(u_x + \epsilon v_x).$$

We defined ζ as,

$$\zeta = KP\tanh\left(\frac{1}{q|u_x|}\right),$$

where, $q = 2K(1 + e^{-2})^{\frac{1}{2}}, e = 2,\ P = P^*h\exp\left[-C(1 - A)\right], C = 20, P^* = 27.5 \times 10^3$ and finally, we get the derivative of the ζ,

$$\begin{aligned}
\zeta_x &= KP_x\tanh\left(\frac{1}{q|u_x|}\right) - \frac{P}{4(1 + e^{-2})^{\frac{3}{2}}|u_x|^3}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)2(1 + e^{-2})u_x u_{xx}, \\
&= KP_x\tanh\left(\frac{1}{q|u_x|}\right) - \frac{PK\,\text{sign}(u_x)}{2EKu_x^2}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)u_{xx}.
\end{aligned}$$

where,

$$P_x = P^*\Big[h_x \exp\left[-C(1-A)\right] + ChA_x \exp\left[-C(1-A)\right]\Big] = P^* \exp\left[-C(1-A)\right]\Big[h_x + ChA_x\Big].$$

Linearizing ζ_x,

$$\begin{aligned}
\zeta_x(u+\epsilon v) &= KP_x \tanh\left(\frac{1}{q|u_x+\epsilon v_x|}\right) - \frac{P\mathsf{sign}(u_x)}{2E(u_x+\epsilon v_x)^2}\operatorname{sech}^2\left(\frac{1}{q|u_x+\epsilon v_x|}\right)(u_{xx}+\epsilon v_{xx}) + \mathsf{o}(\epsilon),\\
&= KP_x\big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{\mathsf{sign}(u_x)}{qu_x^2}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x\big] -\\
&-\frac{P\mathsf{sign}(u_x)}{2E}\left(\frac{1}{u_x^2}-\frac{2\epsilon v_x}{u_x^3}\right)\left(\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{2\mathsf{sign}(u_x)}{qu_x^2}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x\right)\\
&\times(u_{xx}+\epsilon v_{xx}) + \mathsf{o}(\epsilon),\\
&= \zeta_x - \frac{P_x\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_x + \frac{2Pu_{xx}\mathsf{sign}(u_x)\epsilon v_x}{2Eu_x^3}\operatorname{sech}^2\frac{1}{q|u_x|} -\\
&\frac{Pu_{xx}\epsilon v_x}{2E^2Ku_x^4}\tanh\frac{1}{q|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} - \frac{P\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_{xx} + \mathsf{o}(\epsilon),\\
&= \zeta_x - \frac{P_x\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_x + \frac{Pu_{xx}\epsilon v_x}{2E|u_x|u_x^2}\operatorname{sech}^2\frac{1}{q|u_x|} -\\
&\frac{Pu_{xx}\epsilon v_x}{2E^2Ku_x^4}\tanh\frac{1}{q|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} - \frac{P\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_{xx} + \mathsf{o}(\epsilon).
\end{aligned}$$

Then, we can linearize $b(u)$ as,

$$\begin{aligned}
b(u+\epsilon v) &= \mathcal{N}E^2\Big[\zeta_x - \frac{P_x\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_x + \frac{Pu_{xx}\epsilon v_x}{2Eu_x^2|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} -\\
&\frac{Pu_{xx}\epsilon v_x}{2E^2Ku_x^4}\tanh\frac{1}{q|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} - \frac{P\,\mathrm{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_{xx}\Big](u_x+\epsilon v_x) + \mathsf{o}(\epsilon),\\
&= \hat{b}(u) + \mathcal{N}E^2\left[KP_x\tanh\left(\frac{1}{q|u_x|}\right) - \frac{P\mathsf{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right]\epsilon v_x +\\
&-\frac{P_x}{2E|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_x + \frac{Pu_{xx}\epsilon v_x}{2Eu_x|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} -\\
&\frac{Pu_{xx}\epsilon v_x}{2E^2Ku_x^3}\tanh\frac{1}{q|u_x|}\operatorname{sech}^2\frac{1}{q|u_x|} - \frac{P\,\mathrm{sign}(u_x)}{2Eu_x}\operatorname{sech}^2\frac{1}{q|u_x|}\epsilon v_{xx}\Big] + \mathsf{o}(\epsilon),\\
&= \hat{b}(u) + \mathcal{N}E^2KP_x\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{2EK|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)u_{xx}\right]\epsilon v_x\\
&-\frac{\mathcal{N}E^2u_{xx}P}{2E^2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x - \frac{\mathcal{N}E^2P}{2E|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_{xx} + \mathsf{o}(\epsilon).
\end{aligned}$$

Now we linearize $c(u)$,

$$\hat{c}(u+\epsilon v) = \hat{c}(u) - \frac{P\mathcal{N}Eu_{xx}}{2u_x^2}\mathsf{sign}(u_x)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x + \mathcal{N}E^2KP\tanh\left(\frac{1}{q|u_x|}\right)\epsilon v_{xx} + \mathsf{o}(\epsilon)$$

Assuming,

$$M(u) = a(u) - b(u) - c(u)$$

To linearize $M(u)$, we can do the following,

$$M(u + \epsilon v) = M(\hat{u}) + \epsilon \mathcal{L}[u]v + \mathsf{o}(\epsilon)$$

where, $M(\hat{u}) = a(\hat{u}) - b(\hat{u}) - c(\hat{u})$ contains all the non-linear terms and $\mathcal{L}[u]v$ contains the linearized terms. We have,

$$\begin{aligned}
a(u+\epsilon v) - b(u+\epsilon v) - \hat{c}(u+\epsilon v) &= \hat{a}(u) + \Big[2\mathcal{N}_w|u-u_w| + 1\Big]\epsilon v - \hat{b}(u) - \hat{c}(u) \\
&-\mathcal{N}E^2KP_x\Big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{2EK|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big]\epsilon v_x \\
&+\frac{\mathcal{N}E^2u_{xx}P}{2E^2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x + \frac{\mathcal{N}E^2P}{2E|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_{xx} \\
&+\frac{P\mathcal{N}Eu_{xx}}{2u_x^2}\mathsf{sign}(u_x)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\epsilon v_x - \mathcal{N}E^2KP\tanh\left(\frac{1}{q|u_x|}\right)\epsilon v_{xx} + \mathsf{o}(\epsilon) \\
&= \hat{a}(u) - \hat{b}(u) - \hat{c}(u) + \Big[2\mathcal{N}_w|u-u_w| + 1\Big]\epsilon v + \\
&\Big[-\mathcal{N}E^2KP_x\Big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{2EK|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big] + \\
&\frac{\mathcal{N}E^2u_{xx}P}{2E^2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{P\mathcal{N}Eu_{xx}}{2u_x^2}\mathsf{sign}(u_x)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big]\epsilon v_x + \\
&\Big[\frac{\mathcal{N}E^2P}{2E|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) - \mathcal{N}E^2KP\tanh\left(\frac{1}{q|u_x|}\right)\Big]v_{xx} \\
&= \hat{a}(u) - \hat{b}(u) - \hat{c}(u) + \underbrace{\Big[2\mathcal{N}_w|u-u_w| + 1\Big]}_{\alpha}\epsilon v + \Big[-\mathcal{N}E^2KP_x\Big[\tanh\left(\frac{1}{q|u_x|}\right) \\
&-\frac{1}{2EK|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big] + \frac{\mathcal{N}E^2u_{xx}P}{2E^2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \\
&\frac{P\mathcal{N}Eu_{xx}}{2|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big]\epsilon v_x + \underbrace{\Big[\frac{\mathcal{N}E^2P}{2E|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) - \mathcal{N}E^2KP\tanh\left(\frac{1}{q|u_x|}\right)\Big]}_{\gamma}v_{xx} \\
&= M(\hat{u}) + \epsilon\Big[\alpha[u]v + \beta[u]v_x + \gamma[u]v_{xx}\Big] + \mathsf{o}(\epsilon),
\end{aligned}$$

where,

$$\beta[u] = -\mathcal{N}E^2KP_x\Big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big] + \frac{\mathcal{N}u_{xx}P}{2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{P\mathcal{N}Eu_{xx}}{2|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right).$$

Now when, $u_x = 0$,

$$\zeta = KP\tanh\left(\frac{1}{q|u_x|}\right), q = 2K(1+e^{-2}),$$

and

$$\zeta_x = KP_x\tanh\left(\frac{1}{q|u_x|}\right) - \frac{P\mathsf{sign}(u_x)}{2Eu_x^2}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)u_{xx}.$$

Then, $u_x \to \epsilon v_x$,

$$\zeta = KP(1+\mathsf{o}(\epsilon)) = KP + \mathsf{o}(\epsilon),$$

$$\zeta_x = KP_x(1+\mathsf{o}(\epsilon)) = KP_x + \mathsf{o}(\epsilon).$$

Then we have,

$$M(u+\epsilon v) = M(u) + \epsilon(1+\mathcal{N}_w|u-u_w|)v - \mathcal{N}E^2KP_x\epsilon v_x - \mathcal{N}E^2KP\epsilon v_{xx} + \mathsf{o}(\epsilon),$$

where,

$$M(u) = u + \mathcal{N}\Big[\rho_w C_{dw}|u-u_w|(u-u_w) - E^2\zeta_x u_x - E^2\zeta u_{xx}\Big].$$

Thus, in general we have,

$$\mathcal{L}[u]v = \alpha v + \beta v_x + \gamma v_{xx},$$

where,

$$\alpha = 2\frac{\triangle t}{\rho h}\rho_w C_{dw}|u-u_w| + 1, \tag{SIME-1}$$

$$\beta = \begin{cases} -\mathcal{N}E^2KP_x\Big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big] + \\ \frac{\mathcal{N}u_{xx}P}{2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{P\mathcal{N}Eu_{xx}}{2|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right), \text{when } u_x \neq 0, E = (1+e^{-2})^{\frac{1}{2}}, \\ -\frac{\triangle t}{\rho h}K(1+e^{-2})P_x, \text{when } u_x = 0. \end{cases} \tag{SIME-2}$$

$$\gamma = \begin{cases} -\mathcal{N}E^2KP\Big[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\Big], & \text{when } u_x \neq 0, \\ -\frac{\triangle t}{\rho h}KE^2P, & \text{when } u_x = 0. \end{cases} \tag{SIME-3}$$

Now, we need to establish that α, β and γ are smooth function, despite its unusual behavior near the origin and to do so, we can use the following lemma.

Lemma 1. *We can show the following is true.*

- *The n-th derivative of* $\tanh\left(\frac{1}{z}\right)$ *takes the form, for* $n \geq 1$

$$\frac{d^n}{dz^n}\tanh\left(\frac{1}{z}\right) = \frac{1}{z^{2n}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_n} c_{j,n} z^{l_{j,n}} \tanh^{p_{j,n}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,n}}\left(\frac{1}{z}\right)\right] \tag{A}$$

  where $C_n, c_{j,n}$ *are integers (both negative and positive),* m_n *are positive integers,* $l_{j,n}, p_{j,n}$ *and* $q_{j,n}$ *are non-negative integers with* $l_{j,n} < 2n$*,* $1 \leq j \leq n$*.*

- *The function*

$$G(z) = \begin{cases} \tanh(\frac{1}{|z|}) & \text{if } z \neq 0, \\ 1 & \text{if } z = 0. \end{cases} \tag{B}$$

  is infinitely differentiable.

Proof. To prove the first part of the lemma, we will use mathematical induction. For $n = 1$, we have,

$$\frac{d}{dz}\tanh\left(\frac{1}{z}\right) = -\frac{1}{z^2}\cdot\operatorname{sech}^2\left(\frac{1}{z}\right).$$

This is of the form given in equation (A) with $m_1 = 0$, $c_{j,1} = 0$, $l_{j,1} = 0$, $p_{j,1} = 0$, and $q_{j,1} = 0$. Now suppose the formula holds for some $n = k \geq 1$ i.e.,

$$\frac{d^k}{dz^k}\tanh\left(\frac{1}{z}\right) = \frac{1}{z^{2k}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right], \tag{C}$$

We will show that the formula also holds for $n = k + 1$. To do this, we differentiate both sides of the equation (C) with respect to z:

$$\frac{d^{k+1}}{dz^{k+1}}\tanh\left(\frac{1}{z}\right) = \frac{d}{dz}\left(\frac{1}{z^{2k}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right]\right).$$

We first use the product rule and the chain rule. Let,

$$\begin{aligned} f(z) &= \frac{1}{z^{2k}}, \\ g(z) &= \operatorname{sech}^2\left(\frac{1}{z}\right), \\ h(z) &= \sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right). \end{aligned}$$

Then we have

$$\frac{d}{dz}\left(f(z)g(z)h(z)\right) = f'(z)g(z)h(z) + f(z)g'(z)h(z) + f(z)g(z)h'(z).$$

For the first term,

$$\begin{aligned} f'(z)g(z)h(z) &= -\frac{2k}{z^{2k+1}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right], \\ &= -\frac{2k}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}+1} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right]. \end{aligned}$$

and for 2nd term,

$$\begin{aligned} f(z)g'(z)h(z) &= -\frac{1}{z^{2k}}\cdot\frac{2}{z^2}\operatorname{sech}^2\left(\frac{1}{z}\right)\tanh\left(\frac{1}{z}\right)\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right), \\ &= -\frac{2}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right). \end{aligned}$$

and for 3rd term $f(z)g(z)h'(z)$, we need to calculate $h'(z)$:

from,

$$h(z) = \sum_{j=1}^{m_k} c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right).$$

Let's start by examining the general term within the summation:

$$f_{j,k}(z) = c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right).$$

We want to compute d/dz of $f_{j,k}(z)$.

Step 1: Differentiating $z^{l_{j,k}}$, $\frac{d}{dz}(z^{l_{j,k}}) = l_{j,k} z^{l_{j,k}-1}$.

Step 2: Differentiating $\tanh^{p_{j,k}}\left(\frac{1}{z}\right)$,

$$\frac{d}{dz}\left(\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\right) = -\frac{1}{z^2}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^2\left(\frac{1}{z}\right)\cdot p_{j,k}.$$

Step 3: Differentiating $\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)$:

$$\frac{d}{dz}\left(\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right) = \frac{1}{z^2}\tanh\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\cdot q_{j,k}.$$

Now, let's combine the terms we derived:

$$\begin{aligned} c_{j,k} l_{j,k} z^{l_{j,k}-1} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right) - c_{j,k} z^{l_{j,k}}\frac{1}{z^2}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^2\left(\frac{1}{z}\right)\cdot p_{j,k}\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right) \\ + c_{j,k} z^{l_{j,k}} \tanh^{p_{j,k}}\left(\frac{1}{z}\right)\frac{1}{z^2}\tanh\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\cdot q_{j,k} \end{aligned}$$

Further simplification leads to,

$$c_{j,k}l_{j,k}z^{l_{j,k}-1}\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)-p_{j,k}c_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}+2}\left(\frac{1}{z}\right)+$$
$$q_{j,k}c_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right).$$

Finally we have,

$$f(z)g(z)h'(z)=\frac{1}{z^{2k}}\operatorname{sech}^2\left(\frac{1}{z}\right)\sum_{j=1}^{m_k}c_{j,k}l_{j,k}z^{l_{j,k}-1}\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)$$
$$-p_{j,k}c_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}+2}\left(\frac{1}{z}\right)+q_{j,k}c_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right),$$
$$=\frac{1}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\sum_{j=1}^{m_k}c_{j,k}l_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)$$
$$-p_{j,k}c_{j,k}z^{l_{j,k}}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}+2}\left(\frac{1}{z}\right)+q_{j,k}c_{j,k}z^{l_{j,k}}\tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right).$$

Now we can substitute the results to,

$$\frac{d^{k+1}}{dz^{k+1}}\tanh\left(\frac{1}{z}\right)=-\frac{2k}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_k}c_{j,k}z^{l_{j,k}+1}\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)\right]$$
$$-\frac{2}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\sum_{j=1}^{m_k}c_{j,k}z^{l_{j,k}}\tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)+$$
$$\frac{1}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\sum_{j=1}^{m_k}c_{j,k}l_{j,k}z^{l_{j,k}-2}\tanh^{p_{j,k}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right)$$
$$-p_{j,k}c_{j,k}z^{l_{j,k}}\tanh^{p_{j,k}-1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}+2}\left(\frac{1}{z}\right)+q_{j,k}c_{j,k}z^{l_{j,k}}\tanh^{p_{j,k}+1}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k}}\left(\frac{1}{z}\right).$$

which can be written as,

$$\frac{d^{k+1}}{dz^{k+1}}\tanh\left(\frac{1}{z}\right)=\frac{1}{z^{2k+2}}\operatorname{sech}^2\left(\frac{1}{z}\right)\left[\sum_{j=1}^{m_{k+1}}c_{j,k+1}z^{l_{j,k+1}}\tanh^{p_{j,k+1}}\left(\frac{1}{z}\right)\operatorname{sech}^{q_{j,k+1}}\left(\frac{1}{z}\right)\right],$$

where, $c_{j,k+1}$ is derived as a combination of all the $c_{j,k}$ terms and same idea goes for $p_{j,k+1}$ which has the terms of $p_{j,k}$, $q_{j,k+1}$ has the terms of $q_{j,k}$ and $l_{j,k+1}$ has the terms of $l_{j,k}$. Therefore, the formula holds for all $n \geq 1$ by mathematical induction.

Here we show that the real valued function

$$G(z)=\begin{cases}\tanh(|z|^{-1}), & \text{if } z\neq 0\\ 1, & \text{otherwise,}\end{cases} \tag{A1}$$

is infinitely differentiable. It suffices to show that this is true at $z = 0$. Note that $G(z)$ is an even function, so we only need to consider the limit as $z \to 0^+$. We have

$$\lim_{z\to 0^+} G(z) = \lim_{z\to 0^+} \tanh\left(\frac{1}{z}\right) = \lim_{w\to\infty} \tanh(w) = 1.$$

where we make the substitution $w = \frac{1}{z}$ and we have used the fact that $\tanh(w) \to 1$ as $w \to \infty$. Therefore, $\lim_{z\to 0} G(z) = 1$, thus $G(z)$ is continuous.

We have to find $\lim_{z\to 0} \frac{\tanh(|z|^{-1})-1}{z}$ and to do so, we do the following:

$$\begin{aligned}
\lim_{z\to 0} \frac{\tanh(|z|^{-1}) - 1}{z} &= \lim_{z\to 0} \frac{\frac{d}{dz}(\tanh(|z|^{-1}) - 1)}{\frac{d}{dz} z} \\
&= -\lim_{z\to 0} \frac{\operatorname{sech}^2(|z|^{-1})(-|z|^{-2}) - 0}{1} \\
&= -\lim_{z\to 0} \operatorname{sech}^2(|z|^{-1})|z|^{-2} \\
&= 0
\end{aligned}$$

Therefore, $\lim_{z\to 0} \frac{\tanh(|z|^{-1})-1}{z} = 0$. To show that $G^{(n)}(0)$ exists and is finite for all $n \geq 0$,

$$\lim_{z\to 0} \frac{d^n}{dz^n} G(z) = \lim_{z\to 0} \frac{C_n}{z^{2n}} \operatorname{sech}^2\left(\frac{1}{z}\right) \left[\sum_{j=1}^{m_n} c_{j,n} z^{l_{j,n}} \tanh^{p_{j,n}}\left(\frac{1}{z}\right) \operatorname{sech}^{q_{j,n}}\left(\frac{1}{z}\right)\right] = 0,$$

since $\operatorname{sech}^2\left(\frac{1}{z}\right)$ approaches 0 faster than z^{2n} as $z \to 0$, and the sum inside the brackets is bounded as $z \to 0$ by a constant multiple of $z^{l_{1,n}}$, which has degree less than $2n$. Therefore,

$$\lim_{z\to 0} \frac{d^n}{dz^n} G(z) = 0.$$

Using the previous results, we have

$$\begin{aligned}
\lim_{z\to 0} \frac{G^{(n-1)}(z) - G^{(n-1)}(0)}{z} &= \lim_{z\to 0} \frac{1}{z} \cdot \frac{C_{n-1}}{z^{2(n-1)}} \operatorname{sech}^2\left(\frac{1}{z}\right) \left[\sum_{j=1}^{m_{n-1}} c_{j,n-1} z^{l_{j,n-1}} \tanh^{p_{j,n-1}}\left(\frac{1}{z}\right) \operatorname{sech}^{q_{j,n-1}}\left(\frac{1}{z}\right)\right] \\
&= \lim_{z\to 0} \frac{C_{n-1}}{z^{2n-1}} \operatorname{sech}^2\left(\frac{1}{z}\right) \left[\sum_{j=1}^{m_{n-1}} c_{j,n-1} z^{l_{j,n-1}} \tanh^{p_{j,n-1}}\left(\frac{1}{z}\right) \operatorname{sech}^{q_{j,n-1}}\left(\frac{1}{z}\right)\right] \\
&= 0
\end{aligned}$$

Therefore, we have shown that all derivatives of $G(z)$ exist and are continuous at $z = 0$, i.e., $G(z)$ is infinitely differentiable at $z = 0$. □

Remark: By the above lemma, clearly the co-efficients β and γ are C^∞ as $\tanh\frac{1}{|z|}$ and $\operatorname{sech}^2\frac{1}{|z|}$ are C^∞, when extended to $z = 0$ by continuity. Thus, only $\alpha[u]$ is not differentiable at $u = u_w$, but it is continuous.

3.3 Second Order Linear Differential Equation with Discontinuous Coefficient:

Here we concentrate on the linear 2nd order ordinary differential equation with discontinuous coefficient of the form[6],

$$\frac{d^2v}{dx^2} + p(x)\frac{dv}{dx} + q(x)v = r(x). \tag{3.6}$$

In our case, $p = \frac{\beta}{\gamma}, q = \frac{\alpha}{\gamma}$ and $r = \frac{1}{\tau\gamma}\big(r_* - f_n\big)$.

3.3.1 Strong Solution:

In (3.6), We assume that $r(x)$ is integrable function, $q(x)$ and $p(x)$ are absolutely integrable functions throughout the interval (a, b) where $a, b \in \mathbb{R}$.

Definition 1. *We say a function $F(x)$ is absolutely integrable in the interval (a, b) if $\int_a^b |F(x)|\, dx < +\infty$ [6].*

Definition 2. *A function $F(x)$ is integrable throughout $a \leq x \leq b$ if it is continuous except at a finite number of points and if $\int F(x)dx$ converges when extended over any portion of the interval [6].*

Definition 3. *By a solution of (3.6) is understood a function of x which at every point of (a, b) is continuous and has a continuous first derivative and at every point of (a, b) where p, q, r are continuous has a second derivative and satisfies (3.6)[6].*

Theorem 1 (Existence of Solution[6])**.** *If c is any point of the interval (a,b) and Γ, Γ_1 any constants, there exists a solution of (3.6) which satisfies the conditions,*

$$v(c) = \Gamma, v'(c) = \Gamma_1.$$

Proof. Following the steps from [6], we opt to establish the theorem. So, we use the case where p and q vanish at every point of (a, b). Then we have at $x \in (a, b)$,

$$y'' = r(x),$$

and from that, we get the general solution of (3.6),

$$y = \int_c^x \big(\int_c^y r(z)dz\big)dy + A(x-c) + B.$$

Letting $A = \Gamma_1, B = \Gamma$ and integration by parts gives,

$$y = x\int_c^x r(y)dy - \int_c^x yr(y)dy + \Gamma_1(x-c) + \Gamma.$$

Changing notation, we get the following expression,

$$y = \int_c^x (x-\xi)r(\xi)d\xi + \Gamma_1(x-c) + \Gamma. \tag{3.7}$$

Differentiating, we get,

$$y' = \int_c^x r(\xi)d\xi + \Gamma_1, \tag{3.8}$$

and y, y' are absolutely integrable.

Using the method of successive approximations, starting from $y_0 = 0$, we compute the approximations $y_1, y_2, \cdots$ by means of the following equation,

$$\begin{aligned} y_n'' &= -py_{n-1}' - qy_{n-1} + r, \quad n = 1, 2, \cdots \\ y_n(c) &= \Gamma, y_n'(c) = \Gamma_1. \end{aligned} \tag{3.9}$$

Now our aim is to show that y_n approaches a limit as $n \to \infty$, from that we will get the solution of (3.6) whose existence we aim to establish. If we write

$$v_n(x) = y_{n+1}(x) - y_n(x),$$

where,

$$y_n(x) = v_0(x) + v_1(x) + \cdots + v_n(x). \tag{3.10}$$

Now we show that this series converges and represents a solution of (3.6) which satisfies the prescribed conditions. For convenience, we differentiate (3.10),

$$y_n'(x) = v_0'(x) + v_1'(x) + \cdots + v_n'(x). \tag{3.11}$$

Since we commenced our approximation from $y_0 = 0$, hence, from the relation above, for $n = 0$, we have, $v_0(x) = y_1(x) - y_0(x) = y_1(x)$. So, v_0, v_0' are given direclty by (3.7),(3.8). Now we need to find a relation for rest of the terms. We observe that, y_{n+1}, we have,

$$\begin{aligned} y_{n+1}'' &= -py_n' - qy_n + r, \quad n = 1, 2, \cdots \\ y_{n+1}(c) &= \Gamma, y_{n+1}'(c) = \Gamma_1. \end{aligned} \tag{3.12}$$

and subtracting (3.9) from (3.12), we get the following,

$$\begin{aligned} v_n'' &= -pv_{n-1}' - qv_{n-1}, \quad n = 1, 2, \cdots \\ v_n(c) &= 0, v_n'(c) = 0. \end{aligned}$$

Applying (3.7),(3.8),

$$v_n(x) = -\int_c^x (x - \xi)\big(p(\xi)v_{n-1}'(\xi) + q(\xi)v_{n-1}(\xi)\big)d\xi \tag{3.13}$$

and

$$v_n'(x) = -\int_c^x \big(p(\xi)v_{n-1}'(\xi) + q(\xi)v_{n-1}(\xi)\big)d\xi. \tag{3.14}$$

From (3.10) and (3.11), we can conclude that y_n and y_n' are continuous functions of x throughout the interval (a, b).

Let us introduce a positive constant $l \geq 1$ such that,

$$l \geq b - a.$$

We then infer from (3.13),(3.14) that,

$$|v_n(x)| \leq l \int_c^x \big(\, |p(\xi)| + |q(\xi)| \,\big)\big(\, |v_{n-1}'(\xi)| + |v_{n-1}(\xi)| \,\big)\, |d\xi| \tag{3.15}$$

and

$$|v_n'(x)| \leq l \int_c^x \big(\, |p(\xi)| + |q(\xi)| \,\big)\big(\, |v_{n-1}'(\xi)| + |v_{n-1}(\xi)| \,\big)\, |d\xi| \,. \tag{3.16}$$

Since $v_0(x)$ and $v_0'(x)$ are continuous throughout (a, b), there exists a positive constant C such that

$$|v_0(x)| \leq C, |v_0'(x)| \leq C, \tag{3.17}$$

We are now in a position to prove the fundamental inequalities

$$|v_n(x)| \leq C\frac{\left[2l\int_c^x \Big(|p(\xi)| + |q(\xi)| \Big) |d(\xi)| \right]^n}{n!} \tag{3.18}$$

and

$$|v_n'(x)| \leq C\frac{\left[2l\int_c^x \Big(|p(\xi)| + |q(\xi)| \Big) |d(\xi)| \right]^n}{n!}. \tag{3.19}$$

Since (3.18) and (3.19) reduces to (3.17) when $n = 0$, it is sufficient to prove that if it is true when $n = k$, it is true when $n = k + 1$.

Letting $n = k + 1$ in (3.14),(3.15) and replacing $|v_k(\xi)|$ and $|v_k'(\xi)|$ in the (3.16) by values obtained by letting $n = k$ in (3.18),(3.19), we get,

$$|v_{k+1}(x)| \leq C\frac{(2l)^{k+1}}{k!}\int_c^x \big(|p(\xi)| + |q(\xi)| \big) \Big[\int_c^\xi \big(|p(\xi)| + |q(\xi)| \big) |d\xi| \Big]^k |d\xi| \tag{3.20}$$

and

$$\big|v_{k+1}'(x)\big| \leq C\frac{(2l)^{k+1}}{k!}\int_c^x \big(|p(\xi)| + |q(\xi)| \big) \Big[\int_c^\xi \big(|p(\xi)| + |q(\xi)| \big) |d\xi| \Big]^k |d\xi|\,. \tag{3.21}$$

Using the following formula, we can find a special case for (3.18) and (3.19),

$$\int_c^x \phi(\xi)\Big(\int_c^\xi \phi(\xi)d\xi \Big)^k d\xi = \frac{1}{k+1}\Big(\int_c^x \phi(\xi)d\xi \Big)^{k+1}.$$

Letting $M = \int_a^b \big(|p(\xi)| + |q(\xi)| \big)d\xi$, we obtain that special case that we mentioned earlier,

$$|v_n(x)| \leq C\frac{(2lM)^n}{n!} \tag{3.22}$$

and

$$|v_n'(x)| \leq C\frac{(2lM)^n}{n!}. \tag{3.23}$$

(3.22),(3.23) show that at every point of (a, b) the terms of the series (3.10) and (3.11) do not exceed in absolute value the terms of the series,

$$C + C\frac{2lM}{1!} + C\frac{(2lM)^2}{2!} + \cdots,$$

and since this is a convergent series of positive constant terms, it follows by Weierstrass's fundamental test that the series (3.22),(3.23) are uniformly convergent throughout (a, b) and therefore since their terms are continuous, represent continuous functions throughout this interval.

Denoting by $y(x)$ the function represented by (3.10), it is clear that y satisfies the given auxiliary conditions. In order to prove that it is the solution of (3.6) at every point of (a, b) where,p, q, r are continuous.

Let x_0 be any such point, and surround it by an interval say (b, c) lying within (a, b) throughout which p, q, r are continuous. If we multiply (3.11) by $-p(x)$, (3.10) by $-q(x)$ and add the two resulting series together, we get, as we see by referring to the differential equation for u_n,

$$-p(x)y'(x) - q(x)y(x) = u_1''(x) + u_2''(x) + \cdots , \tag{3.24}$$

From the way in which this series was obtained it is clear that it is uniformly convergent throughout (b, c) and that its terms are continuous there. Accordingly, since it may also be obtained by differentiating (3.11) term by term (the first term being omitted) its value is,

$$y''(x) - u_0''(x) = y''(x) - r(x).$$

Substituting this value in (3.24), we see that y satisfies (a, b) throughout (b, c). □

The solution whose existence we have established is unique is stated by the following theorem which we now proceed to prove.

Theorem 2 (Uniqueness of the solution[6])**.** *If two solutions of (3.6) satisfy the conditions* $v(c) = \Gamma, v'(c) = \Gamma_1$*, they are identically equal throughout* (a, b).

The difference of two such solutions is a solution of the homogeneous equation,

$$y''(x) + p(x)y'(x) + q(x)y(x) = 0 \tag{3.25}$$
$$\text{with } y(c) = y'(c) = 0$$

Hence our theorem will be established if we can prove the following more special result: *A solution* y_1 *of (3.25), which satisfies the auxiliary conditions* $y(c) = y'(c) = 0$ *vanishes at every point of (a,b)*.

Proof. Using Sturm method we prove this. Let x_0 be any point of (a, b) and consider a solution y_2 of (3.25) satisfying the auxiliary conditions

$$y_2(x_0) = 0, y_2'(x_0) = 1,$$

where $y_1(x)$ and $y_2(x)$ are the solutions of (3.25) and their Wronskian is

$$W(x) = y_1(x)y_2'(x) - y_2(x)y_1'(x),$$

and it's derivative,

$$W'(x) = y_1(x)y_2''(x) - y_1''(x)y_2(x).$$

Since y_1 and y_2 are solutions of (3.25), we have

$$y_1''(x) = -p(x)y_1'(x) - q(x)y_1(x),$$
$$y_2''(x) = -p(x)y_2'(x) - q(x)y_1(x).$$

Using these $W'(x)$ becomes,

$$W'(x) = -p(x)W(x)$$

which is called *Abel's theorem*.
Solving that we get the *Abel's formula*,

$$W(x) = ke^{-\int p dx},\ k \text{ is a constant.}$$

Since $W(c) = 0$, we see that $k = 0$. Accordingly, $W(x_0) = 0$. But $W(x_0) = y_1(x_0)$. Thus we see that y_1 vanishes at x_0 which was any point of (a, b). □

The Fredholm Alternative is a fundamental result in the theory of linear differential equations. It provides a necessary and sufficient condition for the existence and uniqueness of solutions to a second-order linear differential equation with specified boundary conditions.

Lemma 2. ***Fredhom Alternative:*** *Let $p, q, f \in L^1$, either,*

(I)

$$y'' + p(x)y' + q(x)y = f(x), a < x < b$$

$$\alpha_1 y(a) + \alpha_2 y'(a) = \gamma_1$$

$$\beta_2 y(b) + \beta_2 y'(b) = \gamma_2$$

has a unique solution for all $\alpha_j, \beta_j, \gamma_j, j = 1, 2$.

(II) Or,

$$y'' + p(x)y' + q(x)y = 0, a < x < b$$

$$\alpha_1 y(a) + \alpha_1 y'(a) = \beta_1 y(b) + \beta_2 y'(b) = 0$$

has a non-trivial solution (more than one solution).

Proof. Consider the three IVPs:

1.

$$y_1'' + p(x)y_1' + q(x)y_1 = 0$$

$$y_1(a) = \alpha_2, y_1'(a) = -\alpha_1,$$

2.

$$y_2'' + p(x)y_2' + q(x)y_2 = 0$$

$$y_1(b) = \beta_2, y_2'(b) = -\beta_1,$$

3.

$$y_3'' + p(x)y_3' + q(x)y_3 = f(x)$$

$$y_1(a) = y_1'(a) = 0.$$

By Theorem 1 and Theorem 2 above, the three IVPs above all have a respectively a unique solution, which we denote by y_1, y_2, y_3. We consider a solution for the IVP has the form,

$$y(x) = c_1 y_1(x) + c_2 y_2(x) + c_3 y_3(x),$$

where c_1, c_2 and c_3 are arbitrary constants.

From,

$$\begin{aligned}\alpha_1 y(a) + \alpha_2 y'(a) &= \alpha_1(c_1 y_1(a) + c_2 y_2(a) + \underbrace{c_3 y_3(a)}_{=0}) + \alpha_2(c_1 y_1'(a) + c_2 y_2'(a) + \underbrace{c_3 y_3'(a)}_{=0}) \\ &= \alpha_1(c_1\alpha_2 + c_2 y_2(a)) + \alpha_2(-c_1\alpha_1 + c_2 y_2'(a)) \\ &= c_2(-y_1'(a)y_2(a) + y_1(a)y_2'(a)) = c_2 W[y_1, y_2](a) = \gamma_1,\end{aligned}$$

where,$W[y_1, y_2]$ is the Wronskian of the functions y_1 and y_2. Finally we get,

$$c_2 = \frac{\gamma_1}{W[y_1, y_2](a)}$$

If $\gamma_1 = 0$, then $c_2 = 0$ otherwise y_1, y_2 are linearly independent. Now again from,

$$\begin{aligned}\beta_1 y(b) + \beta_2 y'(b) &= \beta_1(c_1 y_1(b) + c_2 y_2(b) + c_3 y_3(b)) + \beta_2(c_1 y_1'(b) + c_2 y_2'(b) + c_3 y_3'(b)) \\ &= c_1(\beta_1 y_1 + \beta_2 y_1') + c_2(\beta_1 y_2(b) + \beta_2 y_2'(b)) + c_3(\beta_1 y_3(b) + \beta_2 y_3'(b)) \\ &= c_1(-y_2'y_1 + y_1'y_2) + c_2(-y_2'y_2 + y_2'y_2) + c_3(-y_2'y_3 + y_2'y_3),\end{aligned}$$

which is equivalently,

$$\gamma_2 = c_1 W[y_2, y_1](b) + c_3 W[y_2, y_3](b),$$

which can be written as,

$$c_1 = \frac{\gamma_2 - c_3 W[y_2, y_3](b)}{W[y_2, y_1](b)} \text{ if } y_2, y_1 \text{ are linear independent otherwise } c_1 = 0.$$

Now, from the homogeneous problem,

$$y'' + p(x)y' + q(x)y = 0, a < x < b,$$

$$\alpha_1 y(a) + \beta_1 y'(a) = \alpha_2 y(b) + \beta_2 y'(b) = 0,$$

which has the solution of the form,

$$y(x) = c_1 y_1(x) + c_2 y_2(x).$$

Now again,

$$\begin{aligned}\alpha_1 y(a) + \alpha_2 y'(a) &= \alpha_1(c_1 y_1(a) + c_2 y_2(a) + \alpha_2(c_1 y_1'(a) + c_2 y_2'(a)) \\ &= c_2 W[y_1, y_2](a).\end{aligned}$$

from that we get,

$$c_2 = 0.$$

and also,

$$\begin{aligned}\beta_1 y(b) + \beta_2 y'(b) &= \beta_1(c_1 y_1(b) + c_2 y_2(b)) + \alpha_2(c_1 y_1'(b) + c_2 y_2'(b)) \\ &= c_2 W[y_2, y_1](b),\end{aligned}$$

from that we get,

$$c_1 = 0.$$

This completes the proof of the Fredholm alternative for both cases. □

Lemma 3. *If $q(x) \leq 0$, the following BVP*

$$y'' + p(x)y' + q(x)y = f(x), a < x < b$$

$$\alpha_1 y(a) + \beta_1 y'(a) = \alpha_2 y(b) + \beta_2 y'(b) = 0$$

has as its unique solution $y = 0$.

Proof. (Energy Method:)

$$\varphi(x) = \exp(\int p(x)dx).$$

Multiplying the given second-order ODE by $\varphi(x)$ yields:

$$\big(\varphi(x)y'\big)' + q(x)\varphi(x)y = 0.$$

Integrating by parts over the interval (a, b) and using the BVP, we get,

$$\int_a^b \Big(\big(\varphi(x)y'\big)' + q(x)\varphi(x)y(x)\Big)dx = \int_a^b \varphi(x)f(x)dx.$$

Using integration by parts on the left-hand side of the equation, we have:

$$-\int_a^b \varphi(x)(y')^2 dx + [\varphi(x)y']_a^b + \int_a^b q(x)\varphi(x)y(x)dx = \int_a^b \varphi(x)f(x)dx.$$

Since $\alpha_1 y(a) + \beta_1 y'(a) = \alpha_2 y(b) + \beta_2 y'(b) = 0$, the boundary term $[\varphi(x)y']_a^b$ evaluates to zero. We can then rewrite the equation as:

$$-\int_a^b \varphi(x)(y')^2 dx + \int_a^b q(x)\varphi(x)y(x)dx = \int_a^b \varphi(x)f(x)dx.$$

Since $\varphi(x)$ is positive and $q(x) \leq 0$ by assumption, the second integral on the left-hand side is non-positive. Furthermore, the first integral on the left-hand side is also non-positive. Therefore, the sum of these integrals is non-positive.

On the right-hand side, the integral is non-negative since $\varphi(x)$ and $f(x)$ are both non-negative.

Thus, the equation reduces to:

$$-\int_a^b \varphi(y')^2 dx + \int_a^b q(x)\varphi(x)y^2(x)dx = 0.$$

For the integral to be zero, both terms must individually be zero. This implies that $(y')^2 = 0$ and $q(x)\varphi(x)y = 0$. Since $\varphi(x)$ is positive, we have $y' = 0$ and $q(x)y = 0$.

If $q(x) < 0$, then $y = 0$ is the only solution that satisfies $q(x)y = 0$. Therefore, the unique solution to the given boundary value problem is $y = 0$. □

3.4 Weak solutions and finite element method for the linear SIME:

First we will discuss finite element method and then we show the existence of uniqueness of the weak solution.

3.4.1 Finite Element Method

We try to solve the problem with Dirichlet boundary conditions,

$$\begin{aligned} v'' + p(x)v' + q(x)v = r(x), a \leq x \leq b, \\ v(a) = c, v(b) = d. \end{aligned} \tag{3.26}$$

3.4.1.1 Symmetrization

We have,

$$v'' + p(x)v' + q(x)v = r(x).$$

We need to find φ, ψ such that the above equation is equivalent to,

$$(\varphi v')' + \psi v = g. \tag{3.27}$$

We get from (3.27),

$$\varphi' v' + \varphi v'' + \psi v = g.$$

Thus, we require,

$$\varphi' = \varphi p(x),$$
$$\psi = \varphi q(x).$$

From that, we have,

$$\varphi = k e^{\int p(x)dx},$$
$$\psi = \varphi q(x).$$

where, k is a constant and we choose $k = 1$. So, finally we have, $\varphi = e^{\int p(x)dx}, \psi = \varphi q(x)$. So, we have,

$$(e^{\int p(x)dx} v')' + q(x) e^{\int p(x)dx} v = r(x) e^{\int p(x)dx}.$$

which implies,

$$(\varphi v')' + \psi v = g.$$

where, $\varphi = e^{\int p(x)dx}, \psi = \varphi q(x)$ and $g = r(x) e^{\int p(x)dx}$. Observe that we have non-homogeneous boundary conditions for u, this does need to be same for v. By the clever choice of initial guess, we can take boundary conditions for v to be homogeneous. Now we set up the initial guess for our Damped Newton's method.
We claim:-

$$u_0(x) = u^{k-1}(x) - u_z^{k-1}(x) + u_z^k(x)$$
$$\text{and } v_i(a) = v_i(b) = 0.$$

At boundaries,

$$u_i(a) = u(a, t^k),$$
$$u_i(b) = u(b, t^k).$$

Then we construct linear profile:

$$u_z^k(x) = \Big[u^k(b) - u^k(a)\Big](\frac{x}{b-a}) + u^k(a).$$

Using this, we can make sure that, we have a homogeneous boundary conditions for the linear solver at each time step. Finally we have the following problem to solve,

$$(\varphi v')' + \psi v = g, \tag{3.28}$$
$$\text{with } v(a) = v(b) = 0.$$

using the FEM linear solver we find v_{i-1} to update:

$$u_i = u_{i-1} + v_{i-1}.$$

Now coming back to the linear system, we observe that we have to deal with the $\int p(x)dx$ first. And to do so, We use the mid-point integration. Then,we discritize the domain length (L) by dividing into n subintervals. Obtain weak form, by integrating by parts the product of the above equation multiplied by a test function, $w \in H^1(a, b)$ with an additional condition that $w(a) = w(b) = 0$ to solve for $v \in H_0^1(a, b)$,

$$\int_a^b \Big((\varphi(x)v')' + \psi(x, u)v\Big)wdx = \int_a^b g(x, u, u_x, u_{xx}, u^*)wdx.$$

Then integrating by parts, we have,

$$v'\varphi(x)w\Big|_a^b - \int_a^b \varphi(x)v'w'dx + \int_a^b \psi(x, u)vwdx = \int_a^b g(x, u, u_x, u_{xx}, u^*)wdx.$$

Then,we have,

$$-\int_a^b \varphi(x)v'w'dx + \int_a^b \psi(x, u)vwdx = \int_a^b g(x, u, u_x, u_{xx}, u^*)wdx.$$

where the first term has vanished because $w(a) = w(b) = 0$.

Introduce a basis solution and make an anzats

$$v \approx \sum_{i=1}^{N} a_i w_i(x). \tag{3.29}$$

where, $w_i : (a, b) \to \mathbb{R}, i = 1, \cdots, N$ is basis,has the form,

$$w_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{if } x_{i-1} < x < x_i. \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{if } x_i < x < x_{i+1} \\ 0, & \text{Otherwise.} \end{cases}$$

Inserting (4.6) and letting $w = w_j, j = 1, \cdots, N$,we obtain,

$$\int_a^b \varphi(x) w_j'(x) \Big(\sum_{i=1}^{N} a_i w_i'(x) \Big) dx - \int_a^b \psi(x, u) w_j(x) \Big(\sum_{i=1}^{N} a_i w_i(x) \Big) dx = - \int_a^b w_j(x) g(x, u, u_x, u_{xx}, u^*) dx.$$

which can be written as

$$M_{i,j} V_j = b_j,$$

where

$$M_{i,j} = \int_a^b \varphi(x) w_j'(x) w_i'(x) dx - \int_a^b \psi(x, u) w_j(x) w_i(x) dx$$

$$b_j = - \int_a^b w_j(x) g_j(x, u, u_x, u_{xx}, u^*) dx$$

and $V_i = \sum_{i=1}^{N} a_i$ is to be found.

Calculating M and b for FEM:

We introduce, $\int_a^b \varphi(x) w_j'(x) w_i'(x) dx$ and $\int_a^b \psi(x, u) w_j(x) w_i(x) dx$ as S, T respectively. Now we use mid-point quadrature to approximate S, T. And we let $\varphi_{i-\frac{1}{2}}$ and $\varphi_{i+\frac{1}{2}}$ be the mid-point value at $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ respectively.

$$S_{i,i} = \int_{x_{i-1}}^{x_{i+1}} \varphi(x) \big(w_i'(x) \big)^2 dx \approx \frac{\varphi_{i-\frac{1}{2}}}{h_i} + \frac{\varphi_{i+\frac{1}{2}}}{h_{i+1}}, i = 1, \cdots, N$$

$$S_{i,i+1} = \int_{x_i}^{x_{i+1}} \varphi(x) w_i'(x) w_{i+1}'(x) dx \approx - \frac{\varphi_{i+\frac{1}{2}}}{h_{i+1}}, i = 1, \cdots, N$$

$$S_{i,i-1} = \int_{x_{i-1}}^{x_i} \varphi(x) w_i'(x) w_{i-1}'(x) dx \approx - \frac{\varphi_{i-\frac{1}{2}}}{h_i}, i = 1, \cdots, N$$

$$T_{i,i} = \int_{x_{i-1}}^{x_{i+1}} \psi(x,u) w_i(x)^2 dx \approx \frac{1}{3}\left[\psi_{i-\frac{1}{2}} h_i + \psi_{i+\frac{1}{2}} h_{i+1}\right]$$
$$T_{i,i+1} = \int_{x_i}^{x_{i+1}} \psi(x,u) w_i(x) w_{i+1}(x) dx \approx \frac{1}{6}\psi_{i+\frac{1}{2}} h_{i+1}$$
$$T_{i,i-1} = \frac{1}{6}\psi_{i-\frac{1}{2}} h_i$$

Combining all these matrices lead us to the actual matrix,

$$M = S_{i,j} + T_{i,j},\ i = 1, \cdots, N, j = 1, \cdots, N$$

which is a tridiagonal matrix and the entries of the matrix are,

$$M_{i,i} = \frac{\varphi_{i-\frac{1}{2}}}{h_i} + \frac{\varphi_{i+\frac{1}{2}}}{h_{i+1}} - \frac{1}{3}\left[\psi_{i-\frac{1}{2}} h_i + \psi_{i+\frac{1}{2}} h_{i+1}\right], i = 1, \cdots, N,$$
$$M_{i,i+1} = -\frac{\varphi_{i+\frac{1}{2}}}{h_{i+1}} - \psi_{i+\frac{1}{2}} h_{i+1}\left(\frac{1}{6}\right), i = 1, \cdots, N,$$
$$M_{i,i-1} = -\frac{\varphi_{i-\frac{1}{2}}}{h_i} - \psi_{i-\frac{1}{2}} h_i\left(\frac{1}{6}\right), i = 1, \cdots, N$$

Now it's time for load vector formulation.For b we let $g_{j+\frac{1}{2}}$ and $g_{j-\frac{1}{2}}$ be the mid-point value at $[x_j, x_{j+1}]$ and $[x_{j-1}, x_j], j = 1, \cdots, N$.

$$\begin{aligned} b_j = -\int_{x_{j-1}}^{x_{j+1}} g(x,u,u_x,u_{xx},u^*) w_j dx &= -\int_{x_{j-1}}^{x_j} g(x,u,u_x,u_{xx},u^*) w_j dx \\ &\quad -\int_{x_j}^{x_{j+1}} g(x,u,u_x,u_{xx},u^*) w_j dx \\ &\approx -g_{j-\frac{1}{2}}\frac{h_j}{2} - g_{j+\frac{1}{2}}\frac{h_{j+1}}{2} \end{aligned}$$

where, $j = 1, \cdots, N$.

3.4.2 Existence and Uniqueness of Weak solutions:

We call Lax-Milgram Theorem for bounded bi-linear functionals which will guarantee the existence of such a unique solution v. It is first informative to review Riesz Representation Theorem which is an essential outcome for bounded linear functionals.

Theorem 3 (Riesz-Representation[7]). *Let $g : H \to \mathbb{R}$ be a bounded linear functional on H.Then there exists a unique element $v \in H$ such that $g(\varphi) = \langle v, \varphi \rangle$ for all $\varphi \in H$. Moreover, $\|g\| = \|v\|$.*

Theorem 4 (Lax-Milgram[7]). *A bilinear form, say $\langle\langle\cdot,\cdot\rangle\rangle$ on H is a mapping such that $\langle\langle\cdot,\cdot\rangle\rangle : H \times H \to \mathbb{R}$ for which there exists constant $m, n > 0$ such that $\forall v, w \in H$,*

$$\langle\langle v, w\rangle\rangle \leq n \|v\| \|w\|$$

and

$$\langle\langle v, v\rangle\rangle \geq m \|v\|^2$$

Finally, let $g : H \to \mathbb{R}$ be a bounded linear functional on H.Then there exists a unique element $v \in H$ such that

$$\langle\langle v, w\rangle\rangle = \langle g, w\rangle$$

for all $w \in H$.

Proof. We will show that our problem verify the assumptions of Lax-milgram Theorem that would guarantee existence of a unique solution for this problem.
To verify the coercive property,we will use the *Poincaré's inequality* which states that there exists a constant $C > 0$ such that

$$|g|_{L^{2(\Omega)}} \leq C |g'|_{L^{2(\Omega)}},$$

where C depends on $\Omega = (a, b)$ for all $g \in H_0^1(\Omega)$.
We have,

$$\langle\langle v, v\rangle\rangle = \int_\Omega \varphi v'^2 dx - \int_\Omega \psi v^2 dx,$$

and

$$\int_\Omega \psi v^2 dx \leq C \max(q) \int_\Omega \varphi v'^2 dx.$$

Thus,

$$\langle\langle v, v\rangle\rangle \geq \big(1 - C \max(q)\big) \int_\Omega \varphi v'^2 dx \geq \min \varphi \big(1 - C \max(q)\big) \|v\|^2_{H_0^1(\Omega)}.$$

where, $\min \varphi > 0$ since $p(x)$ is absolutely integrable. We get from the above inequality,

$$\langle\langle v, v\rangle\rangle \geq m \|v\|^2_{H_0^1(\Omega)}.$$

where, $m = \min \varphi\big(1 - C\max(q)\big)$and q needs to be bounded above, $q \leq \frac{1}{C}$.
Using *Cauchy- Schwartz* in $L^2(\Omega)$,

$$\langle g, w\rangle = (g, w)_{L^2(\Omega)} \leq \|g\|_{L^2(\Omega)} \|w\|_{L^2(\Omega)} \leq \|g\|_{L^2(\Omega)} \|w\|_V$$

i.e. $\langle g, w\rangle$ is continuous.
Using *Cauchy- Schwartz* in $L^2(\Omega)$,

$$\begin{aligned}\langle\langle v, w\rangle\rangle = \left|\int_\Omega \varphi v'w'dx - \int_\Omega \psi vwdx\right| &\leq \max(p, q)\big(\, \|v'\| \, \|w'\| - \|v\| \, \|w\| \,\big) \\ &\leq n \|v\|_{H^1(\Omega)} \|w\|_{H^1(\Omega)}\end{aligned}$$

is continuous, where $n = \max(p, q)$. □

Lemma 4. *If $p(x)$ and $q(x)$ are bounded, and $r(x)$ is square integrable and also $q(x) \leq \frac{1}{(b-a)^2}$. Then (3.28) has a unique solution in $H_0^1(a, b)$.*

Proof. For $w \in H_0^1(a, b)$, we have the weak of formulation of (3.28),

$$-\int_a^b \varphi(x)v'w'dx + \int_a^b \psi(x)vwdx = \int_a^b g(x)wdx,$$

which can be written formally as,

1. $\langle\langle v, w\rangle\rangle = \langle g, w\rangle$ where, $\langle\langle \cdot, \cdot\rangle\rangle$ is the corresponding bi-linear form and $\langle \cdot, \cdot\rangle$ is the L^2 inner product which defines a linear form on $H_0^1(a, b)$.
   Then we have,

   $$|\langle\langle v, w\rangle\rangle| \leq C_1 \|v'\|_{L^2} \|w'\|_{L^2} + C_2 \|v\|_{L^2} \|w\|_{L^2} \leq C \|v\|_{H^1} \|w\|_{H^1}.$$

   Here,

   $$C_1 = e^{(b-a)\max\limits_{[a,b]}|p(x)|}$$

   $$C_2 = \max_{[a,b]}|q(x)|C_1$$

2. Also,

   $$|\langle g, w\rangle| \leq \|g\|_{L^2} \|w\|_{L^2} \leq \|g\|_{L^2} \|w\|_{H^1}.$$

Thus both the bi-linear and linear forms are continuous on H_0^1.

Now, let,

$$\varphi_0 = \min_{[a,b]} \varphi(x) \geq e^{(b-a)\min\limits_{[a,b]} p(x)}$$

$$\psi_0 = \min_{[a,b]} \psi(x) \geq \min_{[a,b]} (-q(x))\varphi_0 = \max_{[a,b]} (-q(x))\psi_0 = -q_0\varphi_0$$

where, $q_0 = \max\limits_{[a,b]} (-q(x))$.

We have,

$$\begin{aligned}\langle\langle v, v\rangle\rangle &= -\int_a^b \varphi(x)v'(x)v'(x)dx + \int_a^b \psi(x)v(x)v(x)dx \\ &\geq -\varphi_0 \|v'\|_{L^2}^2 + \psi_0 \|v\|_{L^2}^2 .\end{aligned}$$

Moreover, we have, $v \in H_0^1[a,b]$,

$$v(x) = \int_a^x v'(z)dz \implies |v(x)| \leq \int_a^b |v'(z)|dz \implies |v(x)| \leq \sqrt{b-a}\, \|v'\|_{L^2}$$

$$\implies \|v\|_{L^2}^2 \leq (b-a)^2 \|v'\|_{L^2}^2 \text{ (Poincare)}$$

3. Thus,

$$\begin{aligned}\langle\langle v, v\rangle\rangle &\geq \varphi_0(\|v'\|_{L^2}^2 - q_0 \|v\|_{L^2}^2) \\ &\geq \varphi_0(\|v'\|_{L^2}^2 - q_0(b-a)^2 \|v'\|_{L^2}^2) \\ &= \varphi_0(1 - (b-a)^2 q_0) \|v'\|_{L^2}^2 \\ &\geq \alpha_0 \|v\|_{H_0^1}, \quad \alpha_0 = \varphi_0(1-(b-a)^2 q_0)\end{aligned}$$

if $1 - (b-a)^2 q_0 > 0$, i.e. $q_0 < \frac{1}{(b-a)^2}$.

From $(1), (2),$ and (3), we deduce that (3.28) has a unique solution in H_0^1, thanks to Lax-Milgram's Lemma. □

3.4.3 Application to the SIME equation:

Previously, we derived the α, β and γ which are the co-efficents of linear equation and they were defined as,

$$\alpha = 2\frac{\triangle t}{\rho h}\rho_w C_{dw}|u - u_w| + 1, \tag{SIME-1}$$

$$\beta = \begin{cases} -\mathcal{N}E^2KP_x\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right] + \\ \frac{\mathcal{N}u_{xx}P}{2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{P\mathcal{N}Eu_{xx}}{2|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right), \text{when } u_x \neq 0, E = (1+e^{-2})^{\frac{1}{2}}, \\ -\frac{\triangle t}{\rho h}K(1+e^{-2})P_x, \text{when } u_x = 0. \end{cases} \tag{SIME-2}$$

$$\gamma = \begin{cases} -\mathcal{N}E^2KP\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right], & \text{when } u_x \neq 0, \\ -\frac{\triangle t}{\rho h}KE^2P, & \text{when } u_x = 0. \end{cases} \tag{SIME-3}$$

From equation (SIME-3), $\gamma[u]$ satisfies the following lemma.

Lemma 5. *$\gamma[u] \leq -\hat{\delta}$, where $\hat{\delta} > 0$,when $|u_x| \leq K_0$.*

Proof. It suffices to see that

$$g(z) = \frac{\tilde{K}}{|z|}\operatorname{sech}^2\frac{1}{|z|} - \tilde{K}\tanh\frac{1}{|z|}, \text{where, } z = q|u_x|,$$

satisfies, $g(z) \leq 0$; $g(z)$ is increasing on $[0, +\infty)$ and that $g(0) = -1$ and $\lim_{z\to+\infty} g(z) = 0$. For $z \in [0, K_0]$; $g(z) \leq \delta; \delta > 0$, we get, $g(z) = g(K_0)$. □

Theorem 5. *Assume that $\bar{u}(x)$ is sufficiently smooth, then the linearized SIME equation (3.6),*

$$v_{xx} + p(x)v_x + q(x)v = r(x), x \in (a, b)$$
$$\text{with } v(a) = v(b) = 0.$$
$$\text{where, } p(x) = \frac{\beta(x)}{\gamma(x)}, q(x) = \frac{\alpha(x)}{\gamma(x)} \text{ and } r(x) = \frac{r^* - M(\bar{u}(x))}{\gamma(x)}.$$

has a unique solution in H_0^1. Here, $\alpha(x), \beta(x), \gamma(x)$ are functionals of $\bar{u}(x)$,$\bar{u}_x(x)$ and $\bar{u}_{xx}(x)$.

Proof. By lemma 4, it suffices to show that,

(I) $p(x)$ and $q(x)$ are bounded.

To show that $p(x)$ and $q(x)$ are bounded, we need to show that $\beta(\bar{u}(x))$ and $\alpha(\bar{u}(x))$ are bounded for all $x \in (a, b)$. Since $\bar{u}(x)$ is sufficiently smooth, we can assume that $\alpha(\bar{u}(x)), \beta(\bar{u}(x)), \gamma(\bar{u}(x))$ are all bounded for all $x \in (a, b)$. This follows from the fact that α, β, γ are functionals of $\bar{u}_(x)$, $\bar{u}_x(x)$ and $\bar{u}_{xx}(x)$, which are all assumed to be sufficiently smooth. Therefore, $p(x)$ and $q(x)$ are also bounded for all $x \in (a, b)$

(II) $r(x)$ is in $L^2([a, b])$.

since $\gamma(x) < 0$, we have $|\gamma(x)| = -\gamma(x)$ for all $x \in [a, b]$. Thus,

$$\int_a^b |r(x)|^2 dx = \int_a^b \left| \frac{r^* - M(\bar{u}(x))}{-\gamma(x)} \right|^2 dx.$$

Since r^* is a constant and $M(\bar{u}(x))$ is bounded on $[a, b]$, we have

$$\int_a^b |r(x)|^2 dx \leq \frac{(r^*)^2}{\min_{x\in[a,b]}(-\gamma(x))} + \int_a^b |M(\bar{u}(x))|^2 dx < \infty.$$

where the last inequality follows from the fact that $M(\bar{u}(x))$ is bounded on $[a, b]$.

Therefore, $r(x)$ is in $L^2([a, b])$.

(III) $q(x) \leq \frac{1}{(b-a)^2}$ which follows from the fact that for $\triangle t > 0$, $\alpha(x) > 0$ and $\gamma(x) \leq \gamma_0 < 0$.

□

Lemma 6. *If* $\frac{P(x)}{h(x)} \geq c_0 > 0$ *&* $|u_x| < c_1$ *in* $[a, b]$*, then the function,*

$$\gamma(x) = -\frac{\triangle t}{\rho h}(1 + e^{-2})KP\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right]$$

satisfies $\gamma(x) \leq \gamma_0 < 0$ *for some* γ_0 *that depends on* c_1.

Proof. Let $z = \frac{1}{\tilde{q}|u_x|}$, where $\tilde{q}$ is a positive constant. We can rewrite $\gamma(x)$ as

$$\gamma(x) = -\frac{\triangle t}{\rho h}(1 + e^{-2})KP\tilde{\gamma}(z)$$

where $\tilde{\gamma}(z) = \tanh(z) - z\operatorname{sech}^2(z)$.

To establish that $\gamma(x)$ satisfies $\gamma(x) \leq \gamma_0 < 0$, we examine the behavior of $\tilde{\gamma}(z)$.

First, note that $\tilde{\gamma}(z)$ is a decreasing function on the interval $[0, +\infty)$. Additionally, we have $\tilde{\gamma}(0) = 0$.

This implies that for $z \in [z_0, +\infty)$, where $z_0 > 0$, we have $\tilde{\gamma}(z) \leq \tilde{\gamma}(z_0) < 0$.
Now, let's consider the condition $|u_x| < c_1$. In this case, we can choose $z_0 = \frac{1}{\tilde{q}c_1}$. Therefore, for $|u_x| < c_1$, we have $z \in [z_0, +\infty)$, and consequently, $\tilde{\gamma}(z) \leq \tilde{\gamma}(z_0) < 0$.
Finally, we can conclude that $\gamma(x) \leq \gamma_0 < 0$, where $\gamma_0 = -\frac{\triangle t}{\rho h}(1 + e^{-2})c_0\tilde{\gamma}(z_0)$.
In summary, under the assumption $\frac{P(x)}{h(x)} \geq c_0 > 0$ and $|u_x| < c_1$ in $[a, b]$, the function $\gamma(x)$ satisfies $\gamma(x) \leq \gamma_0 < 0$, where γ_0 depends on c_1 and is determined by the expression mentioned above. □

For when, $u_x = 0$, we have,

$$\lim_{u_x \to 0} \beta[u] = -\frac{\triangle t}{\rho h} K E^2 P_x = \beta$$

. And

$$\lim_{u_x \to 0} \gamma[u] = -\frac{\triangle t}{\rho h} K E^2 P = \gamma$$

because,

$$\lim_{z \to 0} \tanh \frac{1}{|z|} = 1; \lim_{z \to 0} \operatorname{sech}^2 \frac{1}{|z|} = 0, z = q|u_x|$$

3.5 Handling the derivatives in the co-efficients:

We observe that the co-efficient of the d such as α, β and γ is formed with the derivative of u. To compute these derivatives, we use Cubic Spline. In this section, we will discuss Cubic Spline.
Consider an interval $\big[a, b\big]$ such that,

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b$$

with $n+1$ nodal points and n sub-intervals. For the function, $y = f(x)$ defined on the same domain gives the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n))\}$. Then the spline interpolation $S(x)$ for $f(x)$ is a piece-wise polynomial function on a sub-interval $\big[x_j, x_{j+1}\big]$ denoted by $S_j(x)$, which satisfy the following conditions:
1) A cubic polynomial $S(x)$ defined by,

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, j = 0, 1, \cdots, n - 1.$$

2) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1}), j = 0, 1, \cdots, n-1$.

3) $S_j(x_{j+1}) = S_{j+1}(x_{j+1}), j = 0, 1, 2, \cdots, n-2$.

4) $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}), j = 0, 1, 2, \cdots, n-2$.

5) $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1}), j = 0, 1, 2, \cdots, n-2$.

6) a) Free (or natural boundary conditions),

$$S''_0(x_0) = 0, S''_{n-1}(x_n) = 0.$$

b) Clamped boundary conditions,

$$S'_0(x_0) = f'(x_0), S'_{n-1}(x_n) = f'(x_n).$$

3.6 Conclusion:

In this chapter, we showed how to linearize SIME equation. Also establish the existence and uniqueness of the solution for the SIME both locally and globally. Lastly we introduced splines by means of which we will calculate the derivative of the solution in the coefficients of the linear equation.

Chapter 4

Verification of the linear solver:

4.1 Algorithm for coding:

Algorithm 1 Using Backward Euler's method on temporal resolution

1: Discritize time $0 = t_0, t_1, \cdots, t_n = T$.

2: $\Delta t = \frac{T}{n}$ on $[0, T]$.

3: While $t < T,$ Call algorithm (2).

Algorithm 2 Using Damped Newton's method

1: With given u_n, tol=10^{-4}.

2: **Process:** solve for v_n from $\mathcal{L}(u_n) \cdot v_n = \frac{1}{\tau}(f - f_n)$(Call: Finite element algorithm).

3: $u_{n+1} = u_n + v_n$.

4: **if** $|u_n - u_{n+1}| <$ tol **then goto** 6.

5: Otherwise, set $u_{n+1} = u_n$ and **goto Process**.

6: Output solution u_{n+1}.

Algorithm 3 Finite element method

1: Symmetrialize the equation.

2: Define Basis solution as piecewise linear equation.

3: initialize domain size, element number, mesh points.

4: Compute $(n-1) \times (n-1)$ matrix (M) and the $(n-1) \times 1$ matrix (b) with entries,

$$M_{jk} = \langle\langle w_j, w_k \rangle\rangle \text{ and } b_k = \langle g, w_k \rangle.$$

5: Solve the linear system of equations,

$$MX = b.$$

6: Get v_n.

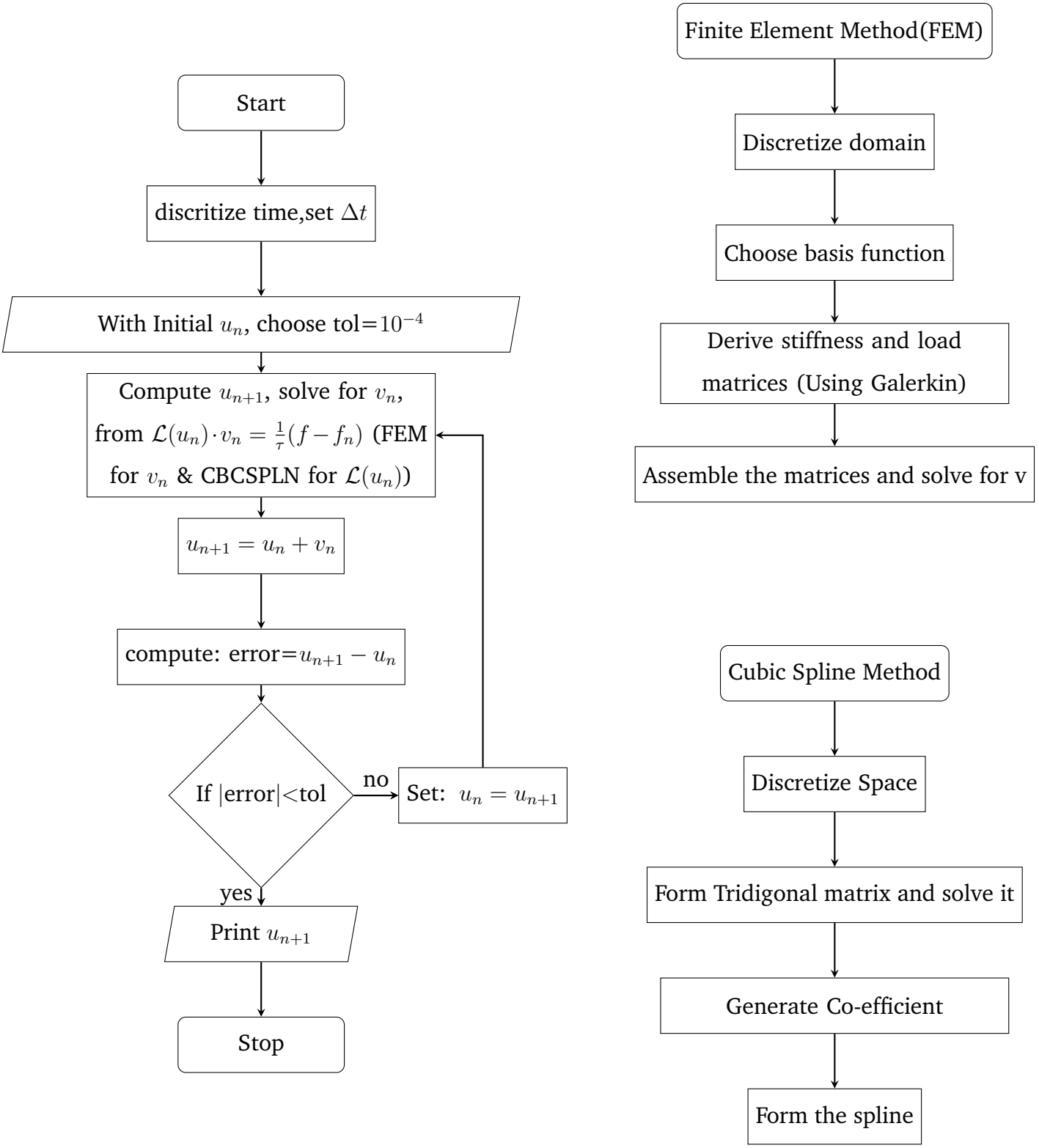


Figure 4.1: Organigram for problem

4.1.1 Validation of the linear solver:

We have the following problem,

$$\frac{d^2v}{dx^2} + p(x)\frac{dv}{dx} + q(x)v = r(x), x \in (a, b)$$
$$v(a) = v(b) = 0.$$

Since we do not have any exact solution to our problem, to validate our code, we set,

$$G(w) \equiv \frac{d^2w}{dx^2} + p(x)\frac{dw}{dx} + q(x)w.$$

We consider two different cases to verify our linear solver (where, the derivative of the solution in the coefficient of the linear equation is computed analytically).

4.1.1.1 Constant co-efficients:

We choose, $p = q = 1$. where, $w = \sin(x)$.

Then, we have on the RHS,

$$\frac{d^2w}{dx^2} + \frac{dw}{dx} + w = -\sin(x) + \cos(x) + \sin(x) = \cos(x).$$

We introduce the integrating factor e^x then the above equation becomes,

$$e^x\frac{d^2v}{dx^2} + e^x\frac{dv}{dx} + e^xv = e^x\cos(x).$$

which can be written as,

$$\left(e^xv'\right)' + e^xv = e^x\cos(x), \tag{A11}$$

with the boundary conditions,

$$v(0) = v(\pi) = 0.$$

First we discritize the domain (π) by dividing into subintervals i.e.

$$(0, \frac{\pi}{4}), (\frac{\pi}{4}, \frac{\pi}{2}), (\frac{\pi}{2}, \frac{3\pi}{4}), (\frac{3\pi}{4}, \pi).$$

Obtain weak form, by integrating by parts the product of the above equation multiplied by a test function, $w \in C_c^\infty(0, \pi)$ with an additional condition that $w(0) = w(\pi) = 0$,

$$\int_0^\pi \Big((e^xv')' + e^xv\Big)wdx = \int_0^\pi we^x\cos(x)dx.$$

Then integrating by parts, we have,

$$v'e^x w\Big|_0^\pi - \int_0^\pi e^x v'w'dx + \int_0^\pi e^x vwdx = \int_0^\pi we^x \cos(x)dx.$$

Then, we have,

$$-\int_0^\pi e^x v'w'dx + \int_0^\pi e^x vwdx = \int_0^\pi we^x \cos(x)dx,$$

where the first term has vanished because $w(0) = w(\pi) = 0$.

Introduce a basis solution and make an anzats

$$v \approx \sum_{j=1}^{3} a_j w_j(x), \tag{4.1}$$

where, $w_j : (a, b) \to \mathbb{R}, j = 1, \cdots, 3$ is basis,has the form,

$$w_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & \text{if } x_{j-1} < x < x_j. \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}, & \text{if } x_j < x < x_{j+1} \\ 0, & \text{Otherwise} \end{cases}$$

Inserting (4.6) and letting $w = w_k, k = 1, \cdots, 3$,we obtain,

$$-\int_0^\pi e^x w_k'(x)\Big(\sum_{j=1}^{3} a_j w_j'(x)\Big)dx + \int_0^\pi e^x w_k(x)\Big(\sum_{j=1}^{3} a_j w_j(x)\Big)dx = \int_0^\pi w_k(x)e^x \cos(x)dx. \tag{4.2}$$

which can be written as

$$MV = b,$$

where,

$$M = -\int_0^\pi e^x w_k'(x)w_j'(x)dx + \int_0^\pi e^x w_k(x)w_j(x)dx,$$

$$b = \int_0^\pi w_k(x)e^x \cos(x)dx,$$

and $V = \sum_{j=1}^{3} a_j$ is to be found.

We call M and b as stiffness matrix and load vector respectively. We name $(w_j', w_k'), (w_j, w_k)$ i.e.$\int_{x_{j-1}}^{x_{j+1}} e^x w_k'(x)w_j'(x)dx$, and $\int_{x_{j-1}}^{x_{j+1}} e^x w_k(x)w_j(x)dx$ as S, T respectively. Now we use mid-point quadrature to approximate S, T.

Table 4.1: The convergence rates with respect to the spatial discretization for (A11).

N	Error	Order
5	0.6267	
10	0.1820	1.98
20	0.0592	2.01
40	0.0201	2.05

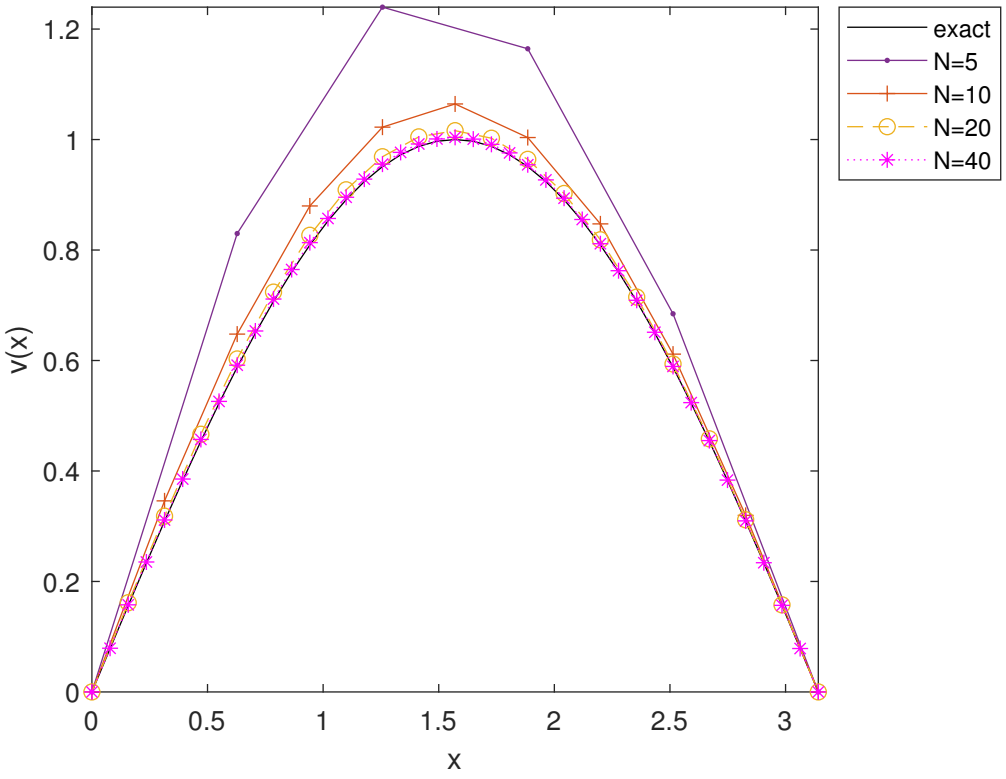


Figure 4.2: Exact solution vs computed solution.

4.1.1.2 Variable co-efficients:

Now we consider the following problem which has variable co-efficients.

$$\frac{d^2v}{dx^2} + p(x)\frac{dv}{dx} + q(x)v = r(x). \tag{4.3}$$

We chose, $p = \sin(x), q = \cos(x)$ and $r = \frac{d^2w}{dx^2} + p(x)\frac{dw}{dx} + q(x)w$, where, $w = \sin(x)$.

Then, we have on the RHS,

$$\frac{d^2w}{dx^2} + p(x)\frac{dw}{dx} + q(x)w = -\sin(x) + \sin(x)\cos(x) + \cos(x)\sin(x) = -\sin(x) + 2\cos(x)\sin(x).$$

(4.3) becomes,

$$\frac{d^2v}{dx^2} + \sin(x)\frac{dv}{dx} + \cos(x)v = -\sin(x) + 2\cos(x)\sin(x). \tag{4.4}$$

We introduce the integrating factor $e^{\int \sin(x)}$ then (4.4) becomes,

$$e^{\int \sin(x)}\frac{d^2v}{dx^2} + e^{\int \sin(x)}\sin(x)\frac{dv}{dx} + e^{\int \sin(x)}\cos(x)v = e^{\int \sin(x)}\big(-\sin(x) + 2\cos(x)\sin(x)\big),$$

which can be written as,

$$\big(\varphi(x)v'\big)' + \psi(x)v = g, \tag{4.5}$$

where, $\varphi = e^{\int \sin(x)}, \psi = \cos(x)\varphi$ and $g = e^{\int \sin(x)}\big(-\sin(x) + 2\cos(x)\sin(x)\big)$.

Finally we get,

$$\begin{aligned} \big(\varphi(x)v'\big)' + \psi(x)v &= g, \\ v(0) = v(\pi) &= 0. \end{aligned} \tag{A12}$$

First we discritize the domain (π) by dividing into subintervals i.e.

$$(0, \frac{\pi}{4}), (\frac{\pi}{4}, \frac{\pi}{2}), (\frac{\pi}{2}, \frac{3\pi}{4}), (\frac{3\pi}{4}, \pi).$$

Obtain weak form, by integrating by parts the product of the above equation multiplied by a test function, $w \in C_c^\infty(0, \pi)$ with an additional condition that $w(0) = w(\pi) = 0$,

$$\int_0^\pi \Big((e^{\int \sin(x)}v')' + \cos(x)e^{\int \sin(x)}v\Big)w dx = \int_0^\pi e^{\int \sin(x)}\big(-\sin(x) + 2\cos(x)\sin(x)\big)dx.$$

Then integrating by parts, we have,

$$v'e^{\int \sin(x)}w\Big|_0^\pi - \int_0^\pi e^{\int \sin(x)}v'w'dx + \int_0^\pi \cos(x)e^{\int \sin(x)}vwdx = \int_0^\pi we^{\int \sin(x)}\big(-\sin(x) + 2\cos(x)\sin(x)\big)dx.$$

Then, we have,

$$-\int_0^\pi e^{\int \sin(x)}v'w'dx + \int_0^\pi \cos(x)e^{\int \sin(x)}vwdx = \int_0^\pi we^{\int \sin(x)}\big(-\sin(x) + 2\cos(x)\sin(x)\big)dx.$$

where, the first term has vanished because $w(0) = w(\pi) = 0$.

Introduce a basis solution and make an anzats

$$v \approx \sum_{j=1}^{3} a_j w_j(x), \tag{4.6}$$

where, $w_j : (a, b) \to \mathbb{R}, j = 1, \cdots, 3$ is basis,has the form,

$$w_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & \text{if } x_{j-1} < x < x_j. \\ \frac{x_{j+1} - x}{x_{j+1} - x_j}, & \text{if } x_j < x < x_{j+1} \\ 0. & \text{Otherwise} \end{cases}$$

Inserting (4.6) and letting $w = w_k, k = 1, \cdots, 3$,we obtain,

$$-\int_0^\pi e^{\int \sin(x)} w_k'(x) \Big(\sum_{j=1}^{3} a_j w_j'(x) \Big) dx + \int_0^\pi \cos(x) e^{\int \sin(x)} w_k(x) \Big(\sum_{j=1}^{3} a_j w_j(x) \Big) dx = \int_0^\pi w_k(x) e^{\int \sin(x)} \big(-\sin(x) + 2\cos(x)\sin(x) \big) dx,$$

which can be written as

$$MV = b.$$

where

$$M = -\int_0^\pi e^{\int \sin(x)} w_k'(x) w_j'(x) dx + \int_0^\pi \cos(x) e^{\int \sin(x)} w_k(x) w_j(x) dx,$$

$$b = \int_0^\pi w_k(x) e^{\int \sin(x)} \big(-\sin(x) + 2\cos(x)\sin(x) \big) dx,$$

and $V = \sum_{j=1}^{3} a_j$ is to be found.

Table 4.2: The convergence rates with respect to the spatial discretization for (A12).

N	Error	Order
5	0.0367	
10	0.0093	1.9754
20	0.0022	2.0844
40	5.2067e-04	2.0801

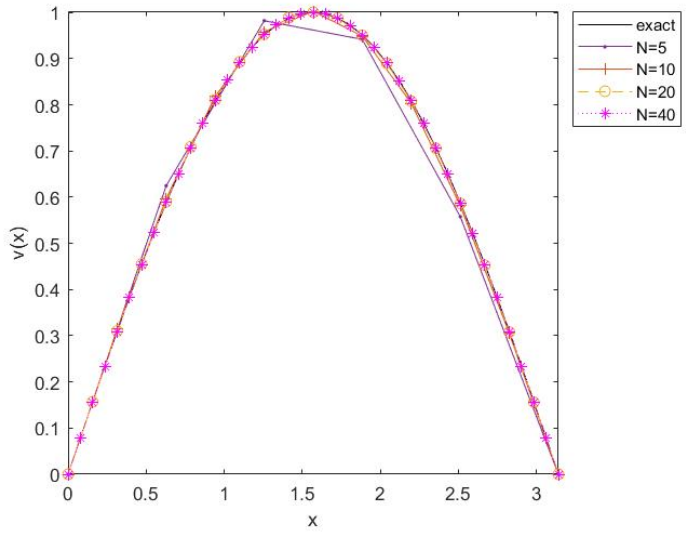


Figure 4.3: Exact solution vs computed solution.

4.2 Validation of the Spline approach to compute derivative:

In this section, we approximate any arbitrary function using Cubic Splines. For the approximation of the function $y = \sin(x)$ using Natural Cubic Spline, we get the following:

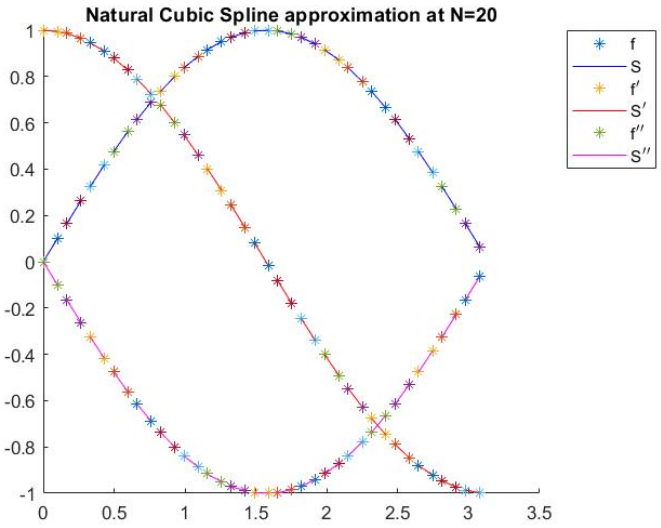
N	error	type	Order
5	4.6146e-04		
10	7.4593e-06	S	5.0867
20	1.7611e-07		5.0134

For the 1st derivative approximation of the function:

N	error	type	Order
5	0.0021		
10	7.8640e-05	S'	4.0507
20	4.1092e-06		3.9502

For the 2nd derivative approximation of the function:

N	error	type	Order
5	0.0370		
10	0.0035	S''	2.9080
20	3.5195e-04		3.0741



For the approximation of the function $y = \sin(x)$ using Clamped Cubic spline:

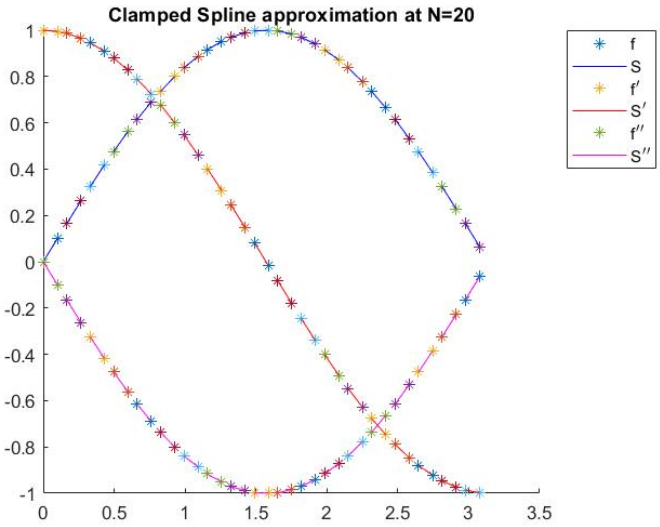
N	error	type	Order
5	1.6404e-04		
10	2.5461e-06	S	5.1368
20	5.9101e-08		5.0361

For the 1st derivative approximation of the function:

N	error	type	Order
5	0.0022		
10	1.0107e-04	S'	3.7986
20	5.2255e-06		3.9644

For the 2nd derivative approximation of the function:

N	error	type	Order
5	0.0341		
10	0.0033	S''	2.8799
20	3.5195e-04		2.9954



4.3 Conclusion:

In this chapter, we introduced our organigram for solving the SIME equation. Also, we verified our linear solver using two different case studies. Lastly, we illustrated the results when we approximate a function and it's derivatives using cubic spline to verify our cubic spline code.

Chapter 5

Testing on a simpler non-linear PDE and Error Analysis:

The aim of this chapter is to solve a partial differential equation with continuous coefficient. We will discretize the temporal resolution and we end with an non-linear ordinary differential equation. To solve this equation, we will use Newton's method and to do so, we linearize the non-linear terms and solve the linear system at each time step using Finite Element methods and we keep updating the solution using Newton's iteration until solution is converged. The derivatives of the of the solution in the coefficients of the linear equation are computed by cubic splines.

Simple Test Case:

To test our algorithm on a simpler problem, we consider the following problem,

$$\begin{aligned} u_t - u_{xx} + \sin(x)u_x + 5\cos(x)|u|^2 = 0, \\ u(0,t) = \sin(t), u(\pi,t) = \sin(\pi + t), \\ u(x,0) = \sin(x). \end{aligned} \tag{5.1}$$

Since we do not have any exact solution to our problem, to validate our code, we set,

$$G(w) \equiv w_t - w_{xx} + \sin(x)w_x + 5\cos(x)|w|^2.$$

We choose, $w(x,t) = \sin(x+t)$.

Then, we have on the RHS,

$$\cos(x+t) + \sin(x+t) + \sin(x)\cos(x+t) + 5\cos(x)|\sin^2(x+t)|.$$

Finally wet our initial boundary problem as below:

$$\begin{aligned} u_t - u_{xx} + \sin(x)u_x + 5|u|^2\cos(x) = \cos(x+t) + \sin(x+t) + \\ \sin(x)\cos(x+t) + 5|\sin^2(x+t)|\cos(x), \\ u(0,t) = \sin(t), u(\pi,t) = \sin(\pi+t), \\ u(x,0) = \sin(x). \end{aligned} \tag{5.2}$$

Handling the time:

Using backward euler method on temporal resolution at times $k = 1, 2, \cdots, K$,
(5.2) becomes,

$$u^k + \triangle t\Big[- u^k_{xx} + \sin(x)u^k_x + 5(|u^2|)^k\cos(x)\Big] = u^{k-1} + \triangle t\Big[\cos(x+k\triangle t) + \sin(x+k\triangle t) + \sin(x)\cos(x+k\triangle t) + 5|\sin^2(x+k\triangle t)|\cos(x)\Big],$$

where, $\dfrac{u^{k-1}}{\triangle t}$ is given solution at a previous time step.

For $k = 1$,

$$u + \triangle t\Big[- u_{xx} + \sin(x)u_x + 5(|u^2|)\cos(x)\Big] = u^* + \triangle t\Big[\cos(x+\triangle t) + \sin(x+\triangle t) + \sin(x)\cos(x+\triangle t) + 5|\sin^2(x+\triangle t)|\cos(x)\Big],$$

where, u^* is given solution at a previous time step.

which can be written as,

$$M(u, u_x, u_{xx}) = G(x, u^*), \tag{5.3}$$

where, $M(u, u_x, u_{xx}) = u + \triangle t\Big[- u_{xx} + \sin(x)u_x + 5(|u^2|)\cos(x)\Big]$,

and, $G(x, u^*) = u^* + \triangle t\Big[\cos(x + \triangle t) + \sin(x + \triangle t) + \sin(x)\cos(x + \triangle t) +$

$5|\sin^2(x + \triangle t)|\cos(x)\Big]$.

Taking care of Non-linear terms:

Consider,

$$\epsilon_1 = |u_x| = \textsf{sign}(u_x).$$

From the LHS of (5.3), we get,

$$M(u + \epsilon v, u_x + \epsilon v_x, u_{xx} + \epsilon v_{xx}) =$$

$$(u + \epsilon v) + \triangle t\Big[- (u_{xx} + \epsilon v_{xx}) + \sin(x)(u_x + \epsilon v_x) + 5\epsilon_1(u + \epsilon v)^2\cos(x)\Big].$$

Simplifying the above equation, we get,

$$u + \triangle t\Big[- u_{xx} + \sin(x)u_x + 5\epsilon_1 u^2\cos(x)\Big] + \epsilon\Big[v + \triangle t\Big[- v_{xx} + \sin(x)v_x + 10uv\epsilon_1\cos(x)\Big]\Big].$$

and further simplification leads to,

$$M(u, u_x, u_{xx}) + \Big[\alpha(x)v_{xx} + \beta(x)v_x + \gamma(x, u)v\Big] + \mathcal{O}(\epsilon),$$

where,

$$\alpha(x) = -\triangle t, \beta(x) = \triangle t\sin(x), \gamma(x, u) = 1 + 10\epsilon_1 u\triangle t\cos(x).$$

Neglecting higher order terms from above and then the differential equation becomes,

$$M(u, u_x, u_{xx}) + \Big[\alpha(x)v_{xx} + \beta(x)v_x + \gamma(x, u)v\Big] = G(x, u^*),$$

which has the form of,

$$v_{xx} + p(x)v_x + \frac{\gamma}{\alpha}v = \frac{G(x, u^*) - M(u, u_x, u_{xx})}{\tau\alpha(x)},$$

where, $p(x) = \frac{\beta}{\alpha}$.

We introduce the integrating factor $e^{\int p(x)dx}$ then the above equation becomes,

$$\left(e^{\int p(x)dx}v'\right)' + e^{\int p(x)dx}\frac{\gamma(x)}{\alpha(x)}v = e^{\int p(x)dx}\left[\frac{G(x,u^*) - M(u,u_x,u_{xx})}{\tau\alpha(x)}\right].$$

The above equation can be written as,

$$\left(\varphi(x)v'\right)' + \psi(x,u)v = g(x), \tag{5.4}$$

where,$\varphi(x) = \alpha(x)e^{\int p(x)dx}, \psi(x,u) = e^{\int p(x)dx}\gamma(x), g(x) = e^{\int p(x)dx}\left[\frac{G(x,u^*) - M(u,u_x,u_{xx})}{\tau}\right],$

with boundary conditions, $v(0) = v(\pi) = 0.$

Solving the linear problem:

Observe that we have non-homogeneous boundary conditions for u, this does need to be same for v. By the clever choice of initial guess, we can take boundary conditions for v to be homogeneous. Now we set up the initial guess for our Damped Newton's method.
We claim:-

$$u_0(x) = u^{k-1}(x) - u_z^{k-1}(x) + u_z^k(x)$$
$$\text{and } v_i(0) = v_i(\pi) = 0.$$

At boundaries,

$$u_i(0) = w(0,t^k) = w^k(0),$$
$$u_i(\pi) = w(\pi,t^k) = w^k(\pi).$$

Then we construct linear profile:

$$u_z^k(x) = \left[w^k(\pi) - w^k(0)\right](\frac{x}{\pi}) + w^k(0).$$

Using this, we can make sure that, we have a homogeneous boundary conditions for the linear solver at each time step. using the FEM linear solver we find v_{i-1} to update:

$$u_i = u_{i-1} + v_{i-1}.$$

5.1 Numerical Experiments:

In our algorithm, we have used the following numerical methods.

a) Backward Euler to handle temporal resolution: we need to check the order of accuracy for this.

b) Newton's method to handle non-linearity: we need to vary the tolerance and damping factor to notice any change in the solution.

c) Finite element method (as a part of Newton's method to solve linear problem): we need to check the order of accuracy.

d) To calculate the coefficients of the FEM, we have used Finite difference method.

5.1.1 Error Estimation:

Consider the following nonlinear differential equation,

$$u_t + u'' + p(x)u' + q(x)u = r(x), , x \in (a,b), t \in [0,T] \tag{5.5}$$
$$u(x,0) = g(x), u(0,t) = u(b,t) = 0$$

has a unique solution,$u(x,t)$.

The solution comes with an error which is composed with,

$$e = e_1 + e_2 + e_3 + e_4 + e_5$$

where, e_1 comes from the discretization of time and e_2 comes from Newton iteration, e_3 comes from integrating factor and e_4 comes from Finite element method to solve linear problem in space and e_5 tolerance of the finite element solver.

We know that in general,

$$e_1 = \mathcal{O}(\triangle t)$$

and

$$e_4 = \mathcal{O}(\triangle x^2)$$

but $e_2, e_5 << e_4$ as tolerance $\approx \epsilon(\triangle x^2)$ and $e_3 << e_4$, so we can neglect them. Now to verify, let $\tilde{p}, \tilde{q}$ and $\tilde{r}$ be finite difference approximation of p, q and r respectively on a grid

with mesh size $h > 0$. Let $\tilde{u}$ be solution of

$$\tilde{u_t} + \tilde{u}'' + \tilde{p}(x)\tilde{u}' + \tilde{q}(x)\tilde{u} = \tilde{r}(x) \tag{5.6}$$

We need to estimate $\|\tilde{u} - u\|_{L_2}$.
Subtracting (5.6) from (5.5),

$$(u_t - \tilde{u_t}) + (u'' - \tilde{u}'') + p(x)u' - \tilde{p}(x)\tilde{u}' + q(x)u - \tilde{q}(x)\tilde{u} = r(x) - \tilde{r}(x) \tag{5.7}$$

Adding some terms to do some adjustments (5.7) can be rewritten as,

$$(u_t - \tilde{u_t}) + (u'' - \tilde{u}'') + p(x)(u' - \tilde{u}') + (p(x) - \tilde{p}(x))\tilde{u}' + q(x)(u - \tilde{u}) + (q(x) - \tilde{q}(x))\tilde{u} = r(x) - \tilde{r}(x) \tag{5.8}$$

Introducing $w = u - \tilde{u}$, (5.8) is reformed as,

$$w_t + w'' + p(x)w' + q(x)w = f$$

where, $f = r(x) - \tilde{r}(x) - (p(x) - \tilde{p}(x))\tilde{u}' - (q(x) - \tilde{q}(x))\tilde{u}$.
Using Backward Euler on temporal resolution,

$$w_{k+1} + \triangle t\big[w''_{k+1} + p(x)w'_{k+1} + q(x)w_{k+1}\big] - w_k - f(t_{k+1})\triangle t = 0$$

The numerical solution of the schema, introduces two error and they are,
a) Global error
b) Local error
Global error is defined as,

$$e_N(\triangle t) = \underbrace{w_K}_{\text{Numerical Solution at } t_K} - \underbrace{w(t_K)}_{\text{Exact Solution at } t_K}$$

where, $t_K = K\triangle t = T$ is fixed as $\triangle t \to 0$. To find the order of e_N, we need to find out local error(residual error).
Let $w(t)$ be an exact solution of the above equation. Then the local error is,

$$e_k(\triangle t) = w(t_{k+1}) + \triangle t\big[w''(t_{k+1}) + p(x)w'(t_{k+1}) + q(x)w(t_{k+1})\big] - w(t_k) - ft(t_{k+1})\triangle t = 0$$

Using Taylor expansion of $w(t_{k+1}) = w(t_k + \triangle t)$ around t_k,

$$e_k(\triangle t) = \big[w(t_k) + w'(t_k)\triangle t + \frac{w''(t_k)}{2}\triangle t^2 + \cdots\big] +$$
$$\triangle t\big[w''(t_{k+1}) + p(x)w'(t_{k+1}) + q(x)w(t_{k+1})\big] - w(t_k) - f(t_{k+1})\triangle t$$
$$= \mathcal{O}(\triangle t^2)$$

We use the following theorem to connect local error with global error.

Theorem 6. *If $e_k(\triangle t) = (\triangle t^{p+1})$, then $e_K(\triangle t) = (\triangle t^p)$. That is, the global error is one order lower than the local error.*

So, in our case,

$$e_1 = \mathcal{O}(\triangle t)$$

i.e Backward euler is a first order method.

Error analysis for finite element methods usually includes two parts:

1) error estimates for V_N space but not for the solution space V. Having said that, we can prove that V_N is the best approximation to the exact solution v.

2) convergence analysis

Theorem 7. *We have,*

1) v_N is the projection of v onto V_N through the inner product $\langle v, \tilde{v}\rangle$ ie.

$$v - v_N \perp V_N \;\; or \;\; v - v_N \perp \tilde{v}_j, j = 1, \cdots, N$$

Then,

$$\langle v - v_N, \bar{v}_N\rangle = 0 \;\; \forall \;\; \bar{v}_N \in V_N \;\; or \;\; \langle v - v_N, \tilde{v}_j\rangle = 0, j = 1, \cdots, N$$

where, $\{\tilde{v}_j\}$'s are the basis functions.

2) v_N is the best approximation in the energy norm, i.e.,

$$\|v - v_N\| \leq \|v - \bar{v}_N\|$$

Now,

$$\begin{aligned}\|v - v_N\|^2 &= \int_a^b \big(-\varphi(x)(v' - v_N')^2 + \psi(x)(v - v_N)^2\big)dx \\ &\leq \varphi_{max}\int_a^b (v' - v_N')^2 dx + \psi_{max}\int_a^b (v - v_N)^2 dx \\ &\leq \max\{\varphi_{max}, \psi_{max}\}\int_a^b \Big((v' - v_N')^2 + (v - v_N)^2\Big)dx \\ &= C\,\|v - v_N\|_1^2\end{aligned}$$

where,$C = \max\{\varphi_{max}, \psi_{max}\}$. Thus, we get,

$$\|v - v_N\| \leq \|v - \bar{v}_N\| \leq \bar{C}\,\|v - \bar{v}_N\|$$

since we are using piece-wise linear space in $H_0^1(a, b)$ spanned by mesh $\{x_j\}, j = 1, \cdots, N.$, we get the following error estimation,

$$\|v - v_N\|_\infty \leq Ch^2\,\|v''\|_\infty$$

.

5.1.2 Derivatives computed by Finite Difference and Integrating factor computed analytically:

The convergency of the method is analyzed for different values of N (length of spatial resolution) and K(length of temporal resolution), selecting maximum number of Newton's iteration is 100 and tolerance is taken to be 10^{-8} for stopping the iterations. . The results are given in table 5.1-5.2. As seen from the table 5.1, while value of N and K increases, the error decreases. But for $N = 80, K = 8000$ ($<$) and 16000, the method starts to decrease convergence rates. However, for $K = 64000$, a better result is observed.

Since we are interested in demonstrating the spacial convergence, we used a fixed K in every cases to keep the temporal error on a negligible level for all considered spacial step sizes.

Table 5.1: The convergence rates with respect to the spatial discretization (varying) and temporal discretization (steady) for (5.4).

N	K	error	Order	K	error	Order	K	error	Order
10	8000	8.1989e-04		16000	8.1448e-04		64000	8.1767e-04	
20	8000	2.2115e-04	1.89	16000	2.1511e-04	1.92	64000	2.1058e-04	1.9572
40	8000	6.5688e-05	1.7513	16000	5.9523e-05	1.8536	64000	5.5034e-05	1.9360
80	8000	2.5965e-05	1.3391	16000	1.9739e-05	1.59	64000	1.5062e-05	1.8694

Through out the all cases, we have observed an oscillatory behavior in the error. Since the right hand side of the equation is highly oscillatory, this sort of behavior is expected. Now, we focus on the temporal resolution. As before,we observe from the table 5.2, while value of K increases, the error decreases and order of accuracy is steadily close to 1.

Table 5.2: The convergence rates with respect to the spatial discretization (Steady) and temporal discretization (Varying) for (5.4) .

N	K	error	Order
160	1000	1.0370e-04	
160	2000	5.3564e-05	0.95
160	4000	2.8484e-05	0.91
160	8000	1.5940e-05	0.8375

5.1.3 Derivatives computed by Finite Difference and Integrating Factor computed numerically:

Now we compute the integrating factor using midpoint method. Taking $K = 64000$ and varying N, we can see that the desired order of accuracy is reached.

Table 5.3: The convergence rates with respect to the spatial discretization (varying) and temporal discretization (steady) for (5.4) .

N	K	error	Order
10	64000	8.1421e-04	
20	64000	2.1055e-04	1.9512
40	64000	5.5037e-05	1.9357
80	64000	1.5064e-05	1.8693

Taking $N = 80$ and varying K, we can see that the desired order of accuracy is reached.

Table 5.4: The convergence rates with respect to the spatial discretization (Steady) and temporal discretization (Varying) for (5.4).

N	K	error	Order
160	1000	1.0420e-04	
160	2000	5.4264e-05	0.96
160	4000	2.8684e-05	0.92
160	8000	1.4540e-05	0.8875

Table 5.5: Order of convergence rates with respect to the tolerance of Newton's method where spatial discretization is steady and temporal discretization is varying for (5.4).

Order of Convergence (N=160)				
Tolerance	Average Newton Iteration	K =1000-2000	K= 2000-4000	K= 4000-8000
10^{-6}	3	0.9956	0.8861	0.8254
10^{-8}	3	0.9531	0.9113	0.8375
10^{-10}	4	0.9529	0.9323	0.8475
10^{-11}	12	0.9529	0.9323	0.8475

Table 5.6: Keeping N= 80, K=4000, tolerance as 10^{-8}, we changed τ to observe the Newton iteration numbers for (5.4)

τ	Number of Newton Iteration	Run time(Checked on an Intel I5 processor with 32GB RAM)
1	3	878.247828 seconds
2	14	877.680062 seconds
3	22	966.117524 seconds
4	30	929.129491 seconds
5	37	898.309183 seconds
10	70	1001.257520 seconds

5.1.4 Derivatives computed by Natural Cubic Spline and Integrating Factor computed numerically:

Table 5.7: (Natural Cubic Spline)The convergence rates with respect to the spatial discretization (changing) and temporal discretization (Steady) .

N	K	error	Order
10	64000	0.0017	
20	64000	4.4394e-04	1.9371
40	64000	1.1491e-04	1.9499
80	64000	3.0230e-05	1.9265

Table 5.8: (Natural Cubic Spline)The convergence rates with respect to the spatial discretization (Steady) and temporal discretization (Varying) .

N	K	error	Order
160	1000	1.0420e-04	
160	2000	5.7377e-05	0.90
160	4000	3.2295e-05	0.82
160	8000	1.9753e-05	0.70

5.1.4.1 Conclusion:

The study explores the performance a nonlinear solver by mean of Newton's method and Finite Element Method and do error estimation for the method numerically. The order of accuracy for the solver is 2nd order in space and 1st order in time. We also observed that the smaller the number of tolerance is Newton's method takes more iteration and damping factor of the newton method need to choose properly as well.

Chapter 6

Performance of the sea ice momentum equation (SIME) solver:

In this chapter, we will discuss the behavior of the proposed numerical approach on the SIME equation. The table showed below has been used to run the simulation.

Table 6.1: Constants and parameters used in the dynamics equations.

C		20	
C_{da}	air drag coefficient	5 $\times 10^{-4}$	
C_{dw}	ocean drag coefficient	0.0055	
e	yield curve axis ratio	2	
ρ	sea ice density	918	kgm^{-3}
ρ_a	constant reference densities for air	1.3	kgm^{-3}
ρ_w	constant reference densities for water	1000	kgm^{-3}
P^*	ice strength parameter	27.5$\times 10^3$	Nm^{-1}
h			m
A			m
u	ice velocity		ms^{-1}
τ_a, τ_w	atmospheric,ocean stresses		Nm^{-2}
η	sea surface elevation		kg/s
k		10^2	
dx	Spatial Resolution	200	km
dt	Temporal Resolution	1.8	$seconds$
L	X-extent	2000	km
T	Final Time	7	$days$

In our study, we chose a length of the domain of $L = 200$ kilometers towards horizontal

direction. We chose the ocean and air stress adopting from [1],

$$u_w = \frac{-0.1(2x - L)}{L};$$

and

$$u_a = 5 + \sin\big(\frac{2\pi t}{\Theta} - 3\big)\sin\big(\frac{\pi x}{L}\big)$$

where, $\Theta = 4.$ days.

6.1 Synthetic Solution:

Since SIME does not have any analytic solution, we have to create our solution by choosing a fixed function following [1], we chose our solution as,

$$w = \frac{1}{10}\sin\left[\left(\frac{4x}{L} - 2\right)^2 + Ct\right],$$

where, $C = 5 \times 10^{-6}$. Using that, we can write our SIME equation as,

$$\rho h u_t + \rho_w C_{dw}|u - u_w|(u - u_w) - \underbrace{(1 + e^{-2})}_{E^2}\zeta_x u_x - \underbrace{(1 + e^{-2})}_{E^2}\zeta u_{xx} = W(x,t),$$

where,

$$W(x,t) = \rho h w_t + \rho_w C_{dw}|w - u_w|(w - u_w) - E^2\zeta_x w_x - E^2\zeta w_{xx},$$

where,

$$\begin{aligned}
w_t &= \frac{c}{10}\cos\left[\big(\frac{4x}{L} - 2\big)^2 + ct\right],\\
w_x &= \frac{8(2x - L)}{5L^2}\cos\left[\big(\frac{4x}{L} - 2\big)^2 + ct\right],\\
w_{xx} &= \frac{16}{5L^2}\cos\left[\big(\frac{4x}{L} - 2\big)^2 + ct\right] - \frac{128}{5L^4}(2x - L)^2\sin\left[\big(\frac{4x}{L} - 2\big)^2 + ct\right],\\
\zeta &= KP\tanh\left(\frac{1}{q|w_x|}\right),\\
\zeta_x &= KP_x\tanh\left(\frac{1}{q|w_x|}\right) - \frac{P\text{sign}(w_x)}{2Ew_x^2}\operatorname{sech}^2\big(\frac{1}{q|w_x|}\big)w_{xx}.
\end{aligned}$$

6.1.1 Initial Conditions:

Initially we chose our ice velocity as,

$$u = \frac{1}{10}\sin\left(\frac{4x}{L} - 2\right)^2,$$

and for ice thickness and area we chose a smoothly varying function defined as,

$$h = 1 + \sin(\frac{\pi x}{L}),$$

and,

$$a = \sin^2(\frac{\pi x}{L}).$$

As previously noted, pressure term is defined as,

$$P = P^* h e^{-C*(1-A)},$$

where, $P^* = 27.5 \times 10^3$ and $C = 20$. We can see the range of values of the functions defined above in the following figure.

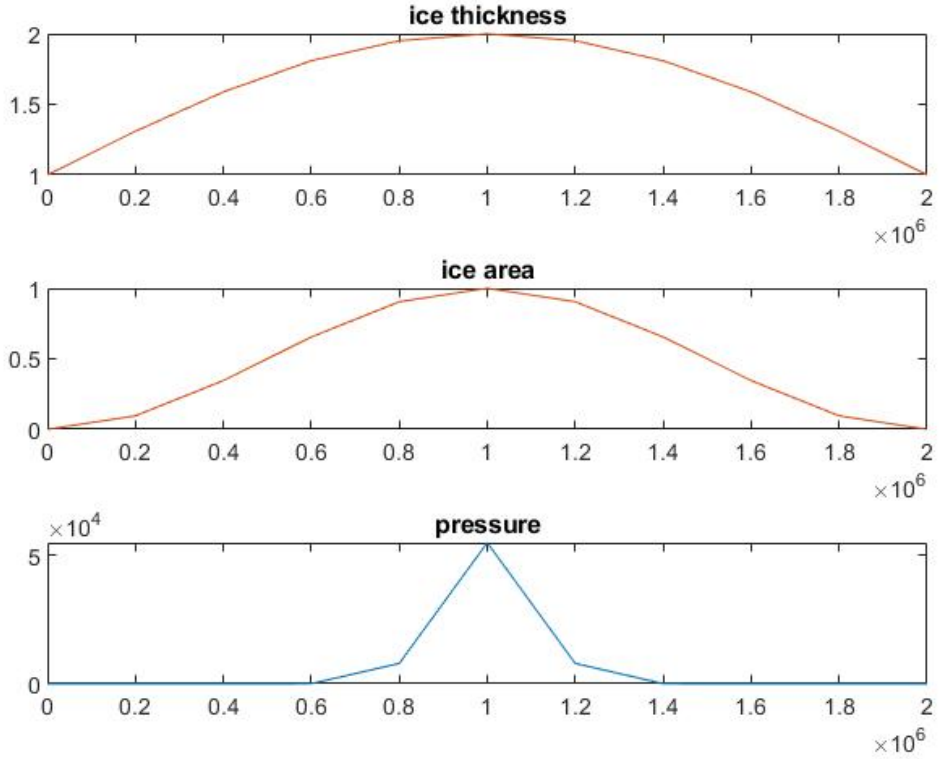


Figure 6.1: Ice thickness, area and pressure plotted against the domain.

6.1.2 Avoiding discontinuity:

Remember,

$$\zeta = KP\tanh\left(\frac{1}{q|w_x|}\right),$$
$$\zeta_x = KP_x\tanh\left(\frac{1}{q|w_x|}\right) - \frac{P\text{sign}(w_x)}{2Ew_x^2}\,\text{sech}^2\,(\frac{1}{q|w_x|})w_{xx}.$$

If we plot them, we can see that,there is discontinuity in the derivative of ζ and same problem arises when we take derivatives of the other terms such as u_x and u_{xx}.

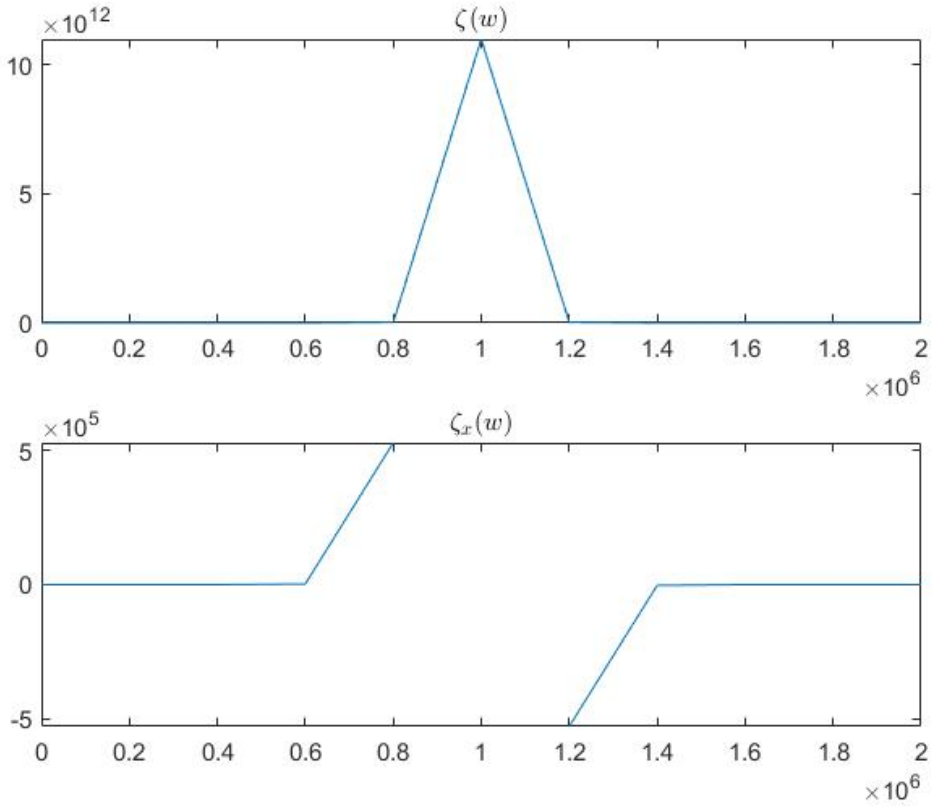


Figure 6.2: The problem of discontinuities.

To overcome this, we added a very small constant namely $\Upsilon = 10^{-10}$ and then we redefine the above terms and similar other derivatives as,

$$\zeta = KP\tanh\left(\frac{1}{q|w_x+\Upsilon|}\right),$$
$$\zeta_x = KP_x\tanh\left(\frac{1}{q|w_x|}\right) - \frac{P\text{sign}(w_x)}{2Ew_x^2}\,\text{sech}^2\,(\frac{1}{q|w_x+\Upsilon|})w_{xx},$$

as a result, we can see from the plot below that the discontinuity problem has been resolved.

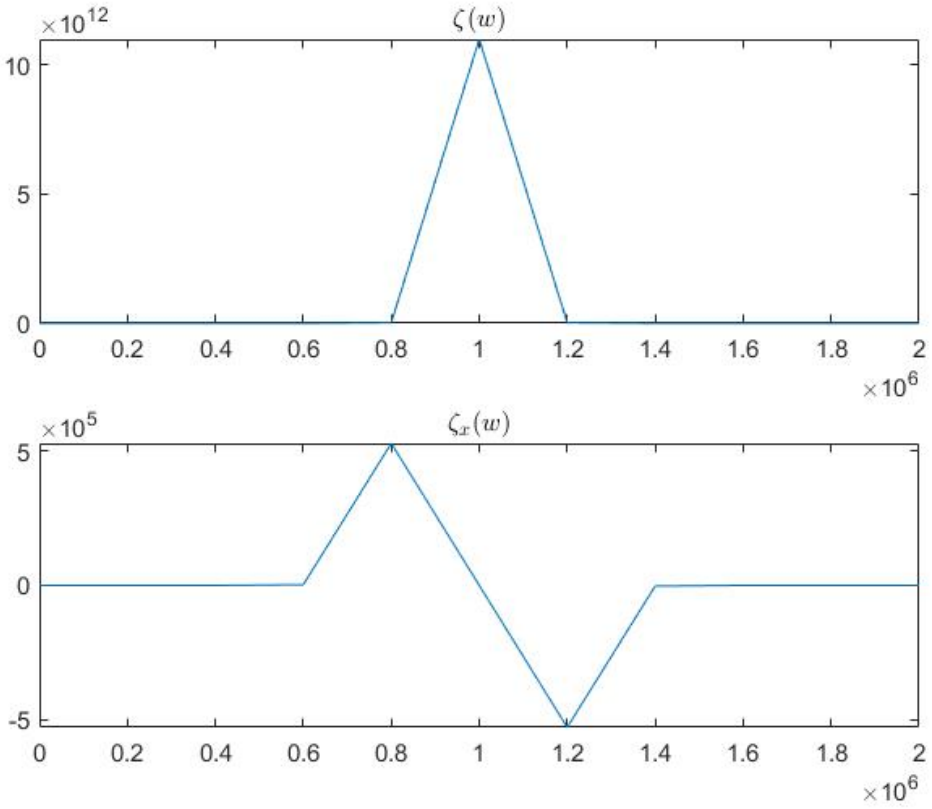


Figure 6.3: Result of adding Υ to the derivatives.

6.1.3 Res-calling the linear problem:

Using Backward Euler on temporal resolution, we get,

$$u^k + \mathcal{N}\left[\rho_w C_{dw}|u^k - u_w|(u^k - u_w) - E^2\zeta_x^k u_x^k - E^2\zeta^k u_{xx}^k\right] = u^{k-1} + \mathcal{N}W(x, k\triangle t),$$

where,

$$\mathcal{N} = \frac{\triangle t}{\rho h}.$$

For $k = 1$, linearizing the above equation,

$$\alpha v + \beta v_x + \gamma v_{xx} = f(x) \tag{6.1}$$

where,

$$f(x) = u^* + \mathcal{N}W(x, \triangle t) - M(u)$$

$$M(u) = u + \mathcal{N}\left[\rho_w C_{dw}|u - u_w|(u - u_w) - E^2\zeta_x u_x - E^2\zeta u_{xx}\right]$$

$$\alpha = 2\frac{\triangle t}{\rho h}\rho_w C_{dw}|u - u_w| + 1$$

$$\beta = \begin{cases} -\mathcal{N}E^2KP_x\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right] + \\ \frac{\mathcal{N}u_{xx}P}{2Ku_x^3}\tanh\left(\frac{1}{q|u_x|}\right)\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right) + \frac{P\mathcal{N}Eu_{xx}}{2|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right), \text{when } u_x \neq 0, E = (1+e^{-2})^{\frac{1}{2}} \\ -\frac{\triangle t}{\rho h}K(1+e^{-2})P_x, \text{when } u_x = 0 \end{cases}$$

$$\gamma = \begin{cases} -\mathcal{N}E^2KP\left[\tanh\left(\frac{1}{q|u_x|}\right) - \frac{1}{q|u_x|}\operatorname{sech}^2\left(\frac{1}{q|u_x|}\right)\right], & \text{when } u_x \neq 0 \\ -\frac{\triangle t}{\rho h}KE^2P, & \text{when } u_x = 0 \end{cases}$$

assuming, $\mathcal{N} = \frac{\triangle t}{\rho h}; E = (1+e^{-2})^{\frac{1}{2}}$.

Let $z = \frac{x}{L} \iff x = Lz$ be the change of variables that re-scales $x \in [0, L]$ to $z \in [0, 1]$. This is also known as a non-dimensionalization procedure.

Let

$$\hat{v}(z) = v(x), \hat{\gamma}(z) = \gamma(x), \hat{\beta}(z) = \beta(x), \hat{\alpha}(z) = \alpha(x), \hat{f}(z) = f(x).$$

We have $\frac{dz}{dx} = \frac{1}{L}$ and $\frac{dx}{dz} = L$. Then,

$$v_x = \frac{dv(x)}{dx} = \frac{d\hat{v}(z)}{dz}\frac{dz}{dx} = \frac{1}{L}\hat{v}_z$$

and

$$v_{xx} = \frac{1}{L^2}\hat{v}_{zz}.$$

Therefore (6.1) is equivalent to

$$\frac{\hat{\gamma}(z)}{L^2}\hat{v}_{zz} + \frac{\hat{\beta}(z)}{L}\hat{v}_z + \hat{\alpha}(z)\hat{v} = \hat{f}(z),\ 0 \leq z \leq 1,\ \ \hat{v}(0) = \hat{v}(1) = 0. \tag{6.2}$$

Dividing over by the coefficient of the second derivative yields

$$\hat{v}_{zz} + p(z)\hat{v}_z + q(z)\hat{v} = r(z),\ 0 \leq z \leq 1,\ \ v(0) = v(1) = 0, \tag{6.3}$$

where

$$p(z) = \frac{L\hat{\beta}(z)}{\hat{\gamma}(z)}, q(z) = \frac{L^2\hat{\alpha}(z)}{\hat{\gamma}(z)}, r(z) = \frac{L^2\hat{f}(z)}{\hat{\gamma}(z)}.$$

Equation 6.3 can be solved using finite elements on the interval $(0, 1)$ (after symmetrization etc.) leading to the approximation of $\hat{v}(z)$ solution of (6.2) which can then mapped back to obtain the solution $v(x) = \hat{v}(z)$ of (6.1).
Introducing the integrating factor, we get,

$$(\varphi\hat{v}')' + \psi\hat{v} = g,$$

where, $\varphi = e^{\int p(z)dz}, \psi = \varphi q(z)$ and $g = r(z)e^{\int p(z)dz}$. So finally with the proper choice of Newton's guess,we have the following problem to solve,

$$(\varphi\hat{v}')' + \psi\hat{v} = g,$$
$$\text{with } \hat{v}(z, 0) = w(z), \hat{v}(0, t) = \hat{v}(1, t) = 0.$$

6.1.4 Stopping criteria:

While calculating the coefficients i.e α, β and γ, it was observed that γ grows exponentially and to fix that we chose that function as,

$$\gamma = \max(\gamma, -10^4),$$

which returns the largest value between each of its compared to -10^4. Also when computing the integrating factor using mid-point integration method,we subtracted the maximum value from it. As in,

$$y = e^{\int p(x)dx},$$

and then,

$$y_1 = y - \max(y).$$

When we form our matrix to solve the linear problem, we find out that the matrix is ill-conditioned as the condition number of the matrix was infinity at ever iteration so traditional matrix solver such as Gauss-seidel method failed to get productive results. So we chose, GMRES (Generalized Minimal Residual) method, the benefit of this method is that it can be restarted, which allows the method to be applied multiple times with different initial guesses for the solution.
We chose separate tolerances for the GMRES method which was 10^{-6} and for the newton iteration was 10^{-8}.

6.1.5 GMRES method:

In MATLAB, GMRES (Generalized Minimal Residual) is a widely used iterative method for solving large, sparse linear systems of equations. It is particularly effective when dealing with ill-conditioned matrices, as it helps overcome the limitations of direct solvers.

The GMRES method is an iterative technique that approximates the solution to a linear system by constructing a sequence of vectors. It aims to minimize the residual, which represents the difference between the left-hand side and the right-hand side of the linear system. The method progressively improves the solution at each iteration until a desired level of accuracy is achieved.

In MATLAB, the gmres function is used to apply the GMRES method to solve linear systems. The basic syntax of the gmres function is as follows:

```
[x, flag, relres] = gmres(A, b,[],tol,N)
```

where, A is the coefficient matrix of the linear system, b is the right-hand side vector of the linear system, x is the computed solution vector, flag is a convergence flag that indicates the success or failure of the iterative process, a value of 0 indicates successful convergence, relres is the relative residual norm, representing the accuracy of the solution. iter is the number of iterations performed, N is the maximum number of iterations, tol is the tolerance level, and [], the initial guess for the solution.

6.1.6 Convergence test:

After doing a convergence test on the solver by varying N and K, we can confirm the 2nd order accuracy for the spatial resolution and 1st order accuracy for the temporal resolution.

Table 6.2: Confirming order of convergence for temporal resolution

N	K	Error	Order
400	24000	1.5957e-08	
400	48000	7.7846e-09	1.0355
400	96000	3.8699e-09	1.0083

Table 6.3: Confirming order of convergence for Spatial resolution

N	K	Error	Order
50	48000	2.2906e-07	
100	48000	4.3949e-08	2.3818
200	48000	7.6884e-09	2.5150

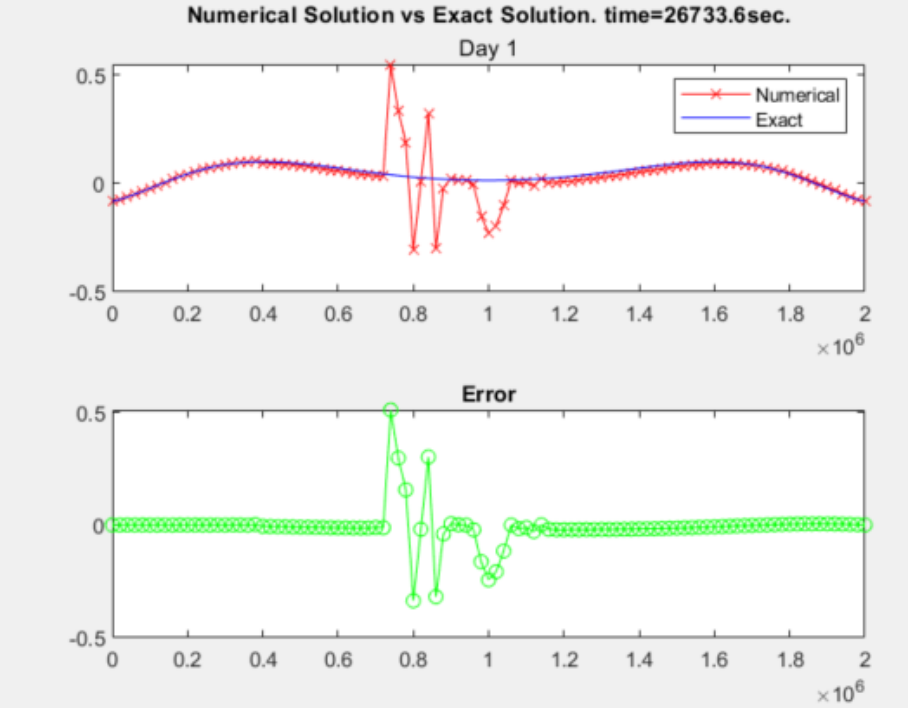


Figure 6.4: Exact solution vs computed solution when $k = 2.5 \times 10^5$.

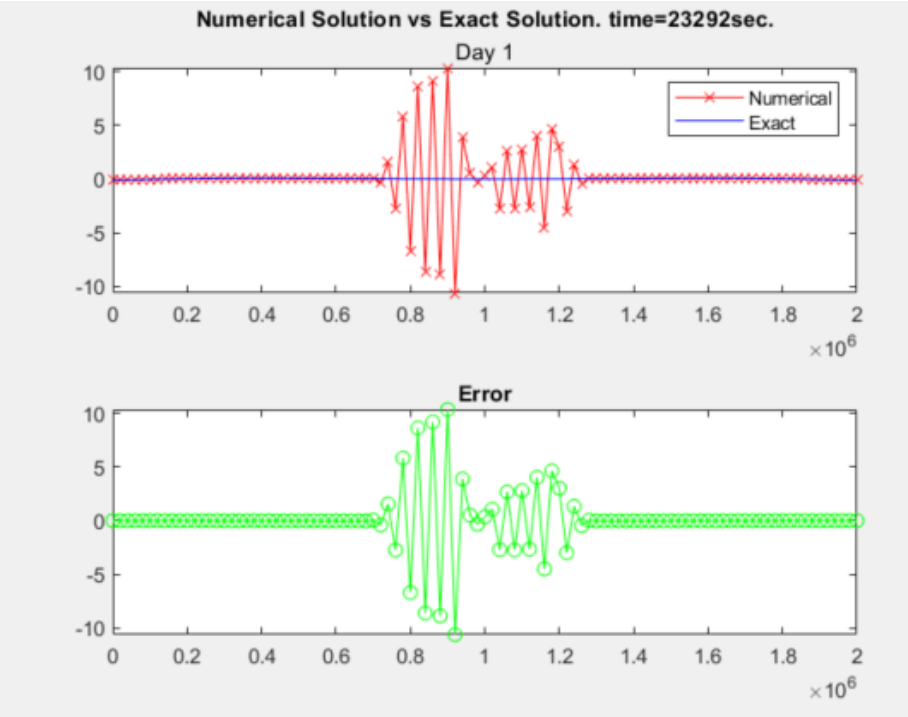


Figure 6.5: Exact solution vs computed solution when $k = 10^3$.

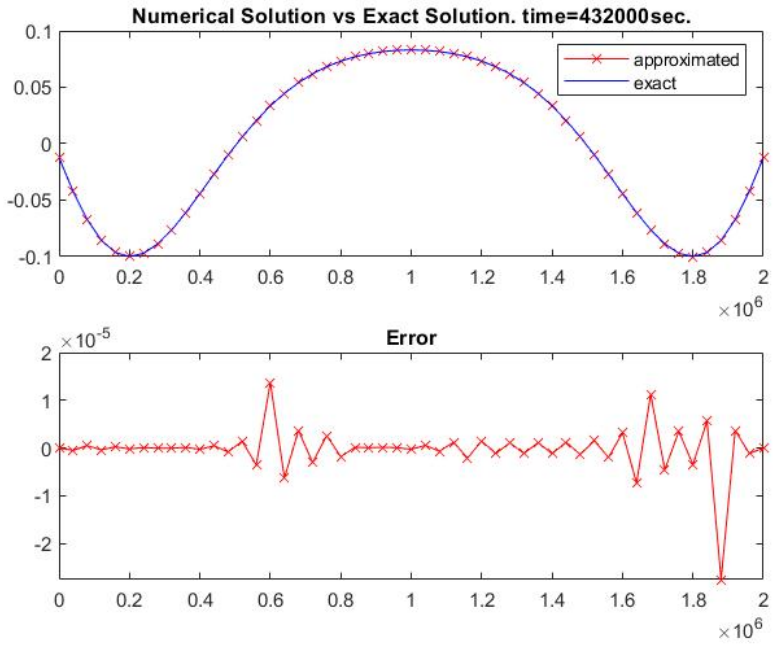


Figure 6.6: Exact solution vs computed solution when $k = 10^2$.

Clearly, $k = 10^2$ is the ideal smoothing constant for the SIME solver.

6.2 Performance of the Nonlinear Solver:

We investigate the performance of the nonlinear solver. Considering different tolerances for the Newton Method, we compute the average number of the iteration and we can find out which tolerance constant is better for our case. To solve the matrix of finite element method, we use GMRES method of the MATLAB.

6.2.1 Convergence of the Newton Solver:

Newton solver refines the initial guess for the solution by updating it based on the residuals of the equation being solved. And the convergence of the solver is evaluated by observing the residual norm, which represents the magnitude of the equation's deviation from zero.

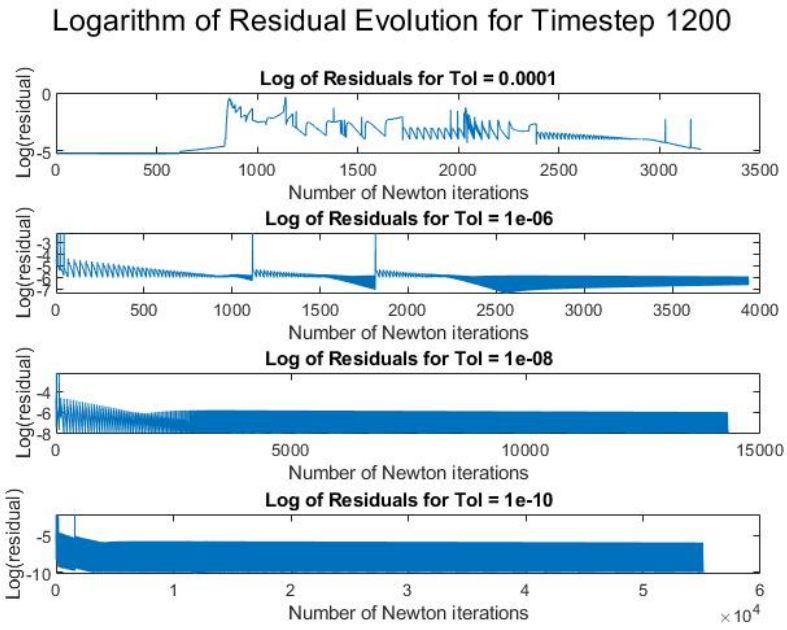


Figure 6.7: Number of Newton Iteration per time step based on tolerances and L2 norm of residue.

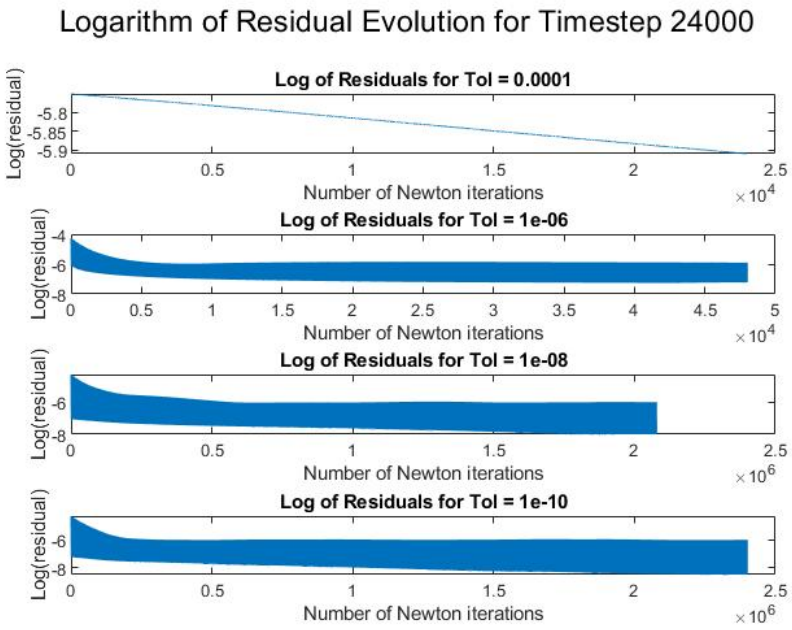


Figure 6.8: Number of Newton Iteration per time step based on tolerances and L2 norm of residue.

6.2.2 Tolerance based:

For this case study, we considered $N = 100$ and $K = 1000.$ and we chose GMRES tolerance as 10^{-4}.

Tolerance	Average Number of Newton Iteration	Number of times Newton Solver failed to converge
10^{-10}	44	93
10^{-8}	10	35
10^{-6}	1	20
10^{-4}	1	3

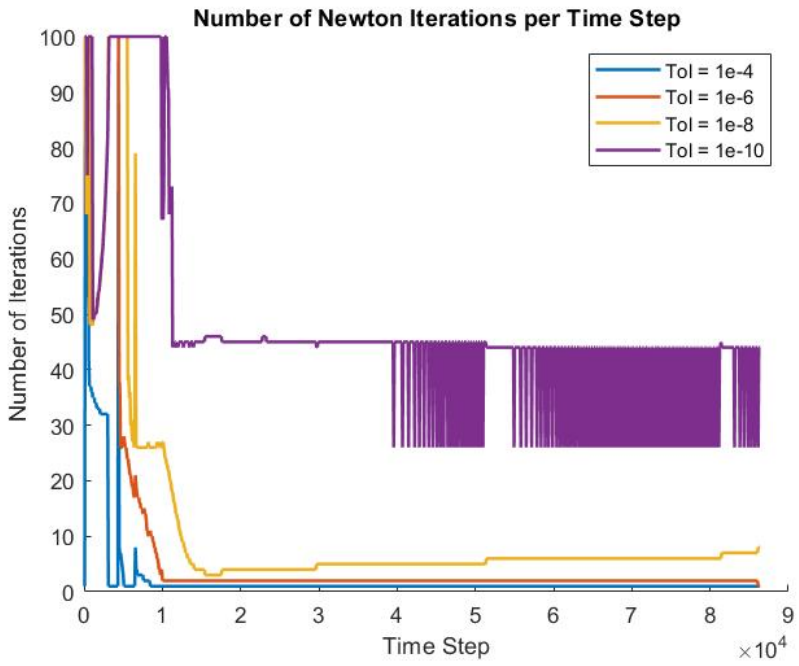


Figure 6.9: Number of Newton Iteration per time step based on Tolerances.

Based on the table above, we can say that 10^{-8} and 10^{-10} is not an ideal tolerance for our case.

6.2.3 Damped Constant τ based:

For this case study, we considered $N = 100$ and $K = 1000.$ and we chose GMRES tolerance as 10^{-4}.

Tau	Average Number of Newton Iteration	Failed to converge	Run time
1	1	1	4673.005371 seconds
2	2	1	3911.647747 seconds
5	7	2	723.709521 seconds
10	22	3	3363.083477 seconds

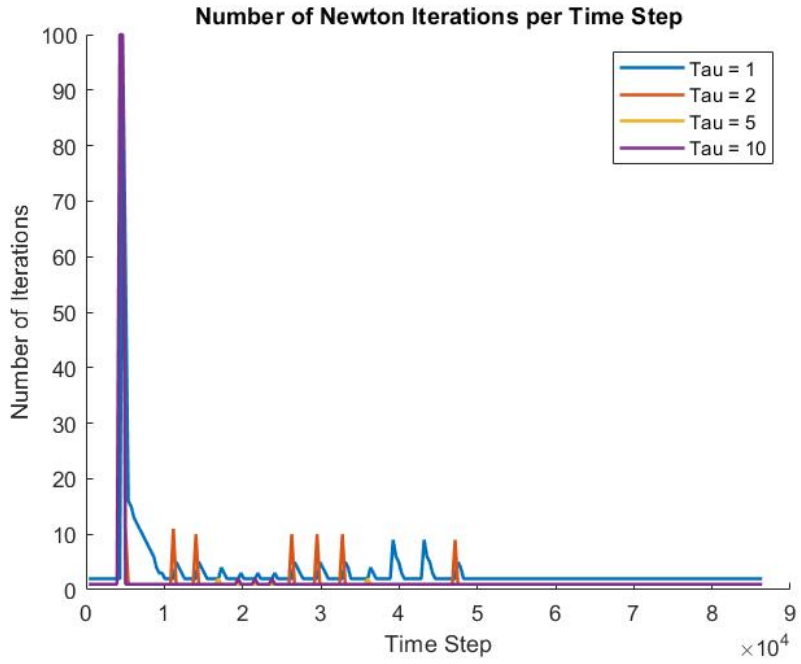


Figure 6.10: Number of Newton Iterations based on damping factors.

Chapter 7

Conclusion

Sea Ice Momentum Equation describes the motion of sea ice under the influence of various forces. It accounts for the advection of momentum by the ice velocity, viscous plastic rheology accounting for internal ice processes, including the pressure gradient driving the ice motion, and external forces acting on the ice namely wind and ocean stress [9]. In our study, we focused on the one-dimensional case, which allowed us to examine how momentum changes along a single direction and also we discussed the possibility of improving the way that the sea ice momentum equation is solved using Newton's method, using a PDE approach of Loeper & Rapetti et al [5].

To apply this method, we linearized the SIME equation and used the Finite Element Method to solve the resulting linear equation at each step. While we tried a damped Newton method, we found that using a unit damping constant worked best for our purposes. The resulting linear equations are difficult to solve because of their ill-conditioned nature. To address this, we used the GMRES method, a numerical technique that helps handle such challenges. As done in Lemieux et al. [10] and Seinen and Khouider [1], we used hyperbolic tangent to smooth the singularity of the viscous plastic sea-ice momentum equation. Because of the ill-conditioning of the mass matrix, however the GMRES worked only when the smoothing constant is set to $K = 100$, while Lemieux et al. [10] and Seinen and Khouider [1] have used $K = 2.5 \times 10^8$. However, we did not fully explore the stability of our approach, which is an important area for future investigation. Also, other possible steps we took to handle these systems such as matrix preconditioning (we rescaled the

whole system), regularization (adding small perturbations to the matrix) and also pivoting (permuting rows or columns of the matrix to ensure that the largest or most significant entries are used during computations).

Additionally, it would be beneficial to extend our analysis to the two-dimensional case. This would provide a more comprehensive understanding of how numerical methods perform and allow us to explore sea ice behavior in a wider range of situations.

In summary, our study highlighted the significance of the sea ice momentum equation in understanding sea ice movement in a one-dimensional setting. Our efforts to improve the numerical solution lay the groundwork for further research. By investigating stability and expanding to the two-dimensional case, we can deepen our knowledge of sea ice dynamics and contribute to the field of sea ice modeling and research.

Bibliography

[1] C.Seinen, B.Khouider.Improving the Jacobian free Newton-Krylov method for the viscous-plastic sea ice momentum equations,Physica D Nonlinear Phenomena,2018, Vol-378,pp. 78-93.

[2] L.P Saumier,M. Agueh and B.Khouider. Optimal Transport for Particle Image Velocimetry,2015,SIAM J. Appl. Math., vol-75,pp-2495–2514.

[3] J.M.N.T Gray.,P.D. Killworth.Stability of the viscous-plastic sea ice rheology.J. Phys. Oceanogr., 25 (1995), pp-971-978.

[4] O.Guba,J.Lorenz,D.Sulsky.On well-posedness of the viscous-plastic sea ice model.J. Phys. Oceanogr., 43 (2013), pp. 2185-2199.

[5] G.Loeper and F.Rapetti. Numerical solution of the Monge-Amp‘ere equation by a Newton's algorithm. C. R. Acad. Sci. Paris, I(340):319–324, 2005.

[6] M.Bocher. Linear Differential Equations with Discontinuous Coefficients, Annals of Mathematics, Second Series, Vol. 6, No. 3 (Apr., 1905), pp. 49-63[logically 97-111]

[7] L.C.Evans, Partial Differential Equations. American Mathematical Society, Providence.

[8] J. Zhang and W.D. Hibler. On an efficient numerical method for modeling sea ice dynamics, J. Geophys. Res.,102(1997)

[9] W. Hibler III. A dynamic thermodynamic sea ice model, J.Phys.Oceanogr.,(1979),pp. 815-846

[10] J.F. Lemieux .,B. Tremblay.,J. Sedlacek,P. Tupper,S. Thomas,D. Huard,J.-P. Auclair. Improving the numerical convergence of viscous-plastic sea ice models with the Jacobian-free Newton-Krylov method J. Comput. Phys., 229 (2010), pp. 2840-2852

[11] M. Losch, S. Danilov. On Solving the Momentum Equations of Dynamic Sea Ice Models with Implicit Solvers and the Elastic-Viscous-Plastic Technique , Ocean Modelling, 41 , pp. 42-52 . doi: 10.1016/j.ocemod.2011.10.002.

[12] C. Mehlmann, T. Richter. A finite element multigrid-framework to solve the sea ice momentum equation. Journal of Computational Physics,Volume 348, Issue CNovember 2017,pp 847–861.

Appendix A: Codes

```
clear all

%% Parameters
lmax=100; % Maximum number of Newton Iterations
Tol=1e-6; % Tolerance Value
K=100;
tau=1;  %damping factor
% initialize variables
v=[];V=[];w=[];phi=[];psi=[];r=[];x=[];
n=100; % number of elements
a=0;
b= 2000000; %[a,b] computational domain. 2000 km
L = (b-a);
h=(b-a)/n;% mesh size 200km
x=a:h:b; %nodal points.
u0=(1/10)*sin((((4*x)/L)-2).^2); % initial condition
Tend=24*60*60; %   1 day
KT=48000;
Dt = Tend/KT;%time step seconds
hei = 1+ sin(pi*x/L); % height
Ar = (sin(x*pi/L)).^2;  % Area
C1 = 5 * 10^(-6);
%%
%linear profile at t = tk
```

```
tk=0;
w_a1 = (1/10)*sin((((4*a)/L)-2).^2 + C1*tk);
w_b1 = (1/10)*sin((((4*b)/L)-2).^2 + C1*tk);
Lk0=(w_b1-w_a1)*(x/L)+ w_a1;
%figure(1),plot(x,Lk0),hold on,title('first solution at t=0')
rho = 918;
N_ = Dt./(rho*hei);
ux1=[];
%% Newton Solver each time step
for k=1:KT
tk=k*Dt;
G1=gfct(x,tk,L,K);% W(x,t)
G = u0 + N_.*G1;
%figure(2),title('W(x,t)'),hold on,plot(x,G);
%linear profile at t = tk
w_a1 = (1/10)*sin((((4*a)/L)-2).^2 + C1*tk);
w_b1 = (1/10)*sin((((4*b)/L)-2).^2 + C1*tk);
Lk=(w_b1-w_a1)*(x/L)+ w_a1;
%form the initial guess
ul0=Lk-Lk0+u0;
%figure(3),plot(x,ul0),hold on
%title('Initial Guess for newton iteration')
for l=1:lmax
% calculate 1st and 2nd derivatives of u
[ux,uxx] = NatSpL(x,ul0); % cubic spline
Ms = Meqn(ul0,x,Dt,ux,uxx,L,K); %M(u)
%figure(5),plot(x,Ms),hold on
%title('Nonlinear u or M(u)')
bts =  bet(x,ux,uxx,Dt,L,K); %beta
%figure(6),plot(x,bts),hold on
%title('Beta function')
gma =  gum(x,ux,Dt,L,K); %gamma
```

```
%figure(7),plot(x,gma),hold on
%title('Gamma function')
als = alph(x,ul0,Dt,L); %alpha
%figure(8), plot(x,als), hold on, title('Alpha function')
L1=L;
fx = (L1^2*(G-Ms))./(tau*gma);
%tau=max(1,max(abs(fx)/100));
%fx = fx/tau;
%figure(9),plot(x,fx),hold on, title('fx'),
%%% integrating factor
paa = (L1*bts)./gma;
qaa = (L1^2*als)./gma;
%figure(10), plot(x,paa),hold on, title('p(x)')
%figure(11), plot(x,qaa),hold on, title('q(x)')
% Rectangle Method
h1=h/L1;
y1(1)= paa(1)*h1/2;
for ks = 2:length(x)
y1(ks) = y1(ks-1) + h1*paa(ks);
end
y1=y1-max((y1));
%figure(221)
%plot(x,y1)
%pause
rx=  exp(y1).*fx; %r
%plot(x,rx,'o-'), title('r(x)')
phix=  exp(y1);
%figure(111),plot(x,phix,'o-'), hold on, title('\phi(x)')
psix=  exp(y1).*qaa;
%figure(112),plot(x,psix,'o-'), title('\psi(x)')
%assemble mass matrix and right hand side
[M,B] = matrix_assemb(phix,psix,rx,h1);
```

```
%codM=condest(M)
[V,FLAG,RELRES]  = gmres(M,B,[],1.e-4,50);
if(FLAG>0)
display(['FLAG=', num2str(FLAG),'GMRES RELRES=',num2str(RELRES)])
end
%figure(21),plot(V), title('V')
%get solusion at nodal points.
v(1)=0;v(n+1)=0;
v(2:n)=V;
%update ul
ul0=ul0+v;
%convergence test
errNwtn = mean(abs(v)) ;
if(errNwtn<Tol)
break
end
end
ux1=[ux1;ux(1)];
u0=ul0;
Lk0=Lk;
figure(33)
subplot(2,1,1)
plot(x,u0,'rx-'),hold on
plot(x,(1/10)*sin((((4*x)/L)-2).^2 + C1*tk),'b-'), hold off
legend('Numerical','Exact');
title(['Numerical Solution vs Exact Solution. time=',num2str(tk),
   'sec.'])
subtitle('Day 1')
subplot(2,1,2)
plot(x,u0-(1/10)*sin((((4*x)/L)-2).^2 + C1*tk),'go-'),%hold on
title('Error')
end
```

```
errorq=sqrt(mean((1/10)*sin((((4*x)/L)-2).^2 + C1*tk)-u0).^2)
```

```
    function [M,b] = matrix_assemb(phi,psi,r,h)

    n=length(r)-1;
    phi_hlf=(phi(1:n)+phi(2:n+1))/2;
    psi_hlf=(psi(1:n)+psi(2:n+1))/2;
    r_hlf=(r(1:n)+r(2:n+1))/2;

    M=sparse(n-1,n-1);b=zeros(n-1,1);

    for I=1:n-2
    M(I,I) = (phi_hlf(I)+phi_hlf(I+1))/h -
       h*(psi_hlf(I)+psi_hlf(I+1))/3;
    M(I+1,I) = -phi_hlf(I+1)/h - h*psi_hlf(I+1)/6;
    M(I,I+1) = M(I+1,I);
    b(I)= -(r_hlf(I)+r_hlf(I+1))*h/2;
    end

    M(n-1,n-1) = (phi_hlf(n-1)+phi_hlf(n))/h -
       h*(psi_hlf(n-1)+psi_hlf(n))/3;
    b(n-1)= -(r_hlf(n-1)+r_hlf(n))*h/2;
```

```
function [yy,yyy] = NatSpL(x,y)

n =  length(x);
for i=1:n
a(i) = y(i);
end
for i=1:n-1
h(i)=x(i+1)-x(i);
end

```

```
for i=2:n-1
alfa(i)=3/h(i)*(a(i+1)-a(i))-3/h(i-1)*(a(i)-a(i-1));
end

l(1)=1;
mu(1)=0;
z(1)=0;

for i=2:n-1
l(i)=2*(x(i+1)-x(i-1))-h(i-1)*mu(i-1);
mu(i)=h(i)/l(i);
z(i)=(alfa(i)-h(i-1)*z(i-1))/l(i);
end

l(n)=1;
z(n)=0;
c(n)=0;

for i=n-1:-1:1
c(i)=z(i)-mu(i)*c(i+1);
b(i)=(a(i+1)-a(i))/h(i)-h(i)*(c(i+1)+2*c(i))/3;
d(i)=(c(i+1)-c(i))/(3*h(i));
end

for i=1:n-1
yy(i)=b(i);
yyy(i)=2*c(i);
end
yy(n)=b(n-1)+2*c(n-1)*h(n-1)+3*d(n-1)*h(n-1)^2;
yyy(n)=2*c(n-1)+6*d(n-1)*h(n-1);
```

```
    function Ms = Meqn(u,x,Dt,ux,uxx,L,K)
```

```
rho = 918; % ice density
hei = 1+ sin(pi*x/L); % height
N1 = Dt./(rho*hei);
%
rho_w = 1026; %kg m^-3
C_dw = 0.0055;
u_w = (-0.1*(2*x-L))./L;
%
e = 2;
erp = 1+ e^(-2);
E = erp^(1/2);
%
%K =  2.5*10^8;
%pr = 1;
pr = 27.5*10^3; % pressure const
Ar = (sin(x*pi/L)).^2;  % Area
Ar_d = 2*(pi/L).*sin(x*pi/L).*cos(x*pi/L); % Area der
C = 20;
hei_d = (pi/L)*cos(x*pi/L);
ypr = pr.*hei.*exp(-C*(1-Ar)); % Pressure
ypr_d = pr.*exp(-C*(1-Ar)).*(hei_d+ C.*hei.*Ar_d); % pressure der
%
%figure(34),clf
%subplot(4,1,1), plot(x,hei_d)
%subplot(4,1,2), plot(x,Ar_d)
%subplot(4,1,3), plot(x,ypr)
%subplot(4,1,4), plot(x,ypr_d)

%pause

%
n1=length(x);
```

```
    qa = 2*K*E;
    for I=1:n1
    thta =  1./(qa*abs(ux(I))+1e-10);
    if(qa*abs(ux(I))>10e-10)
    zet(I) = (K.*ypr(I).*tanh(thta));
    zet_d (I) =
       (K*ypr_d(I).*tanh(thta)-sign(ux(I)).*(ypr(I)./(2*E.*ux(I).^2))...
    .*(sech(thta)).^2.*uxx(I));%
    else
    zet(I) = K.*ypr(I);
    zet_d (I) = K*ypr_d(I);
    end
    end
    %figure(35),clf
    %subplot(3,1,1), plot(x,thta)
    %subplot(3,1,2), plot(x,zet)
    %subplot(3,1,3), plot(x,zet_d)
    %pause

    Ms= u+
       N1.*(rho_w*C_dw.*abs(u-u_w).*(u-u_w)-erp.*zet_d.*ux-erp.*zet.*uxx);
    end
```

```
    function gma =  gum(x,ux,Dt,L,K)

    rho = 918;
    %
    e = 2;
    erp = (1+e^-2);
    E = erp^(1/2);
    %
    pr = 27.5*10^3; % pressure const
```

```
    %pr = 1;
    hei = 1+ sin(pi*x/L); % height
    Ar = (sin(x*pi/L)).^2;  % Area
    C = 20;
    ypr = pr.*hei.*exp(-C*(1-Ar)); % Pressure
    %
    qa = 2*K*E;
    thta =  1./(qa*abs(ux)+1e-10);
    Nq = Dt./(rho.*hei);

    n1=length(x);
    for I=1:n1
    if((qa*abs(ux(I))>1e-10))
    gma(I) = (-Nq(I)*erp*K.*ypr(I).*(tanh(thta(I))-
       (1./(qa*abs(ux(I))+1e-10)).*(sech(thta(I)).^2)));
    else
    gma(I) = (-Nq(I)*K*erp.*ypr(I));
    end
    end

    gma = max(gma,-1e4);

    %figure(111),hold on
    %plot(x,gma)
    %pause
    end
```

```
    function G=gfct(x,t,L,K)
    % this is the nonlinear PDE of the form of known solution
    C1 = 5 * 10^(-6);
    %display(['C1=',num2str(C1)])
    %display(['time in gfct, tk = ',num2str(t/60)])
```

```
% known solution w and its derivatives
w = (1/10)*sin((((4*x)/L)-2).^2 + C1*t);
wt = (C1/10)*cos((((4*x)/L)-2).^2 + C1*t);
wx = (8/(5*L^2))*(2*x-L).*cos((((4*x)/L)-2).^2 + C1*t);

wxx = (16/(5*L^2)).*cos((((4*x)/L)-2).^2 +
   C1*t)-(8*16/(5*L^2)).*(2*x/L-1).^2.*sin((((4*x)/L)-2).^2 +
   C1*t);
%
rho = 918; % ice density
hei = 1+ sin(pi*x/L); % height
%
rho_w = 1026; %kg m^-3
C_dw = 0.0055;
u_w = (-0.1*(2*x-L))./L;
%
e = 2;
erp = 1+ e^(-2);
E = erp^(1/2);
%
pr = 27.5*10^3; % pressure const
%pr = 1;
Ar = (sin(x*pi/L)).^2;  % Area
Ar_d = 2*(pi/L).*sin(x*pi/L).*cos((x*pi/L)); % Area der
C = 20;
hei_d = (pi/L)*cos(x*pi/L);
ypr = pr.*hei.*exp(-C*(1-Ar)); % Pressure
ypr_d = pr.*exp(-C*(1-Ar)).*(hei_d+ C.*hei.*Ar_d); % pressure der
%
qa = 2*K*E;
n1=length(x);
ux=wx;
```

```
    uxx=wxx;
    for I=1:n1
    thta =  1./(qa*abs(ux(I))+1e-10);
    if(qa*abs(ux(I))>10e-10)
    zetw(I) = (K.*ypr(I).*tanh(thta));
    zetw_d (I) =  (K*ypr_d(I).*tanh(thta)-sign(ux(I)).*(ypr(I)./...
    (2*E.*ux(I).^2)).*(sech(thta)).^2.*uxx(I));%
    else
    zetw(I) = K.*ypr(I);
    zetw_d (I) = K*ypr_d(I);
    end
    end

    G= rho.*hei.*wt+
       rho_w*C_dw.*abs(w-u_w).*(w-u_w)-erp.*zetw_d.*wx-erp.*zetw.*wxx;
    %figure(12), plot(x,G),title(['Time=',num2str(t)])

    end
```

```
    function r=GaussSeidel_sp(T,rhs)
    % Use Gauss-Seidel iterative method to solve x for Mx=b;
    [n,r]=size(T);
    if n~=r | n<0
    error('A must be a square matrix');
    end
    Toler = 1e-4; % Set precision
    MaxIter = 10000; % Set maximum iterations
    Iter = 1;
    r0 = zeros(n,1); % Initial value
    r = zeros(n,1); % output x
    while Iter < MaxIter
    for j=1:n
```

```
    r(j) =(-T(j,1:j-1)*r(1:j-1)-T(j,j+1:n)*r0(j+1:n)+rhs(j))/T(j,j);
    end
    if norm(r-r0)<Toler
    break;
    end
    r0=r;
    Iter = Iter+1;
    end
    %disp(['Number of Iterations:' num2str(Iter)]);
    if Iter >= MaxIter
    disp('Maximum Number of Iterations exceeded!');
    end
```

```
    function bts =  bet(x,ux,uxx,Dt,L,K)
    rho = 918;
    hei = 1+ sin(pi*x/L); % height
    hei_d = (pi/L)*cos(x*pi/L);
    Ar = (sin(x*pi/L)).^2;  % Area
    Ar_d = 2*(pi/L).*sin(x*pi/L).*cos((x*pi/L)); % Area der
    N1 = Dt./(rho*hei);

    %
    e = 2;
    erp = 1+ e^(-2);
    E = erp^(1/2);
    %

    pr = 27.5*10^3; % pressure const
    %pr = 1;
    C = 20;
    ypr = pr.*hei.*exp(-C*(1-Ar)); % Pressure
    ypr_d = pr.*exp(-C*(1-Ar)).*(hei_d+ C.*hei.*Ar_d); % pressure der
```

```
    %
    qa = 2*K*E;
    thta =  1./(qa*abs(ux)+1e-10);

    n1=length(x);
    for I=1:n1
    if(qa*abs(ux)>1e-10)
    bts(I) = (-N1(I)*erp*K.*ypr_d(I).*(tanh(thta(I))-...
    (1./(qa*abs(ux(I))+1e-10).*(sech(thta(I)).^2)))+...
    (N1(I).*uxx(I).*ypr(I))./(2*K.*ux(I).^3).*tanh(thta(I))..
    .*(sech(thta(I)).^2) + (ypr(I).*N1(I).*E.*uxx(I))/...
    2.*(qa*abs(ux(I))+1e-10).*(sech(thta(I)).^2));
    else
    bts(I) =  (-N1(I).*K*erp.*ypr_d(I));%.*(qa*abs(ux)<=1e-10);
    end
    end
```

```
    function als = alph(x,u,dt,L)

    rho_w = 1026;
    C_dw = 0.0055;
    rho = 918;
    %K=2.5e8;
    hei = 1+ sin(pi*x/L); % height
    u_w = -0.1*(2*x-L)/L;
    N_w = (dt./(rho.*hei))*rho_w*C_dw;
    als = 2*N_w.*abs(u-u_w)+1;
    %als = (2*N_w.*abs(u-u_w)+1)/K;
    end
```