

AMERICAN UNIVERSITY OF BEIRUT

NEWTON TYPE METHODS FOR SOLVING
FINITE-ELEMENT SPACE,
EULER-IMPLICIT TIME DISCRETIZATION
OF THE HASEGAWA-MIMA EQUATION

by
MAYA ADEL ASSIDI

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Mathematics
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
September 2021

AMERICAN UNIVERSITY OF BEIRUT

NEWTON TYPE METHODS FOR SOLVING
FINITE-ELEMENT SPACE,
EULER-IMPLICIT TIME DISCRETIZATION OF
THE HASEGAWA-MIMA EQUATION

by

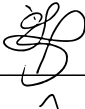
MAYA ADEL ASSIDI

Approved by:

[Signature] 


Dr. Nabil Nassif, Professor
Department of Mathematics

Advisor

[Signature] 

Dr. Sophie Moufawad, Assistant Professor
Department of Mathematics

Co-Advisor

[Signature] 

Dr. Ghassan Antar, Professor
Department of Physics

Member of
Committee

[Signature] 

Dr. Ahmad Sabra, Assistant Professor
Department of Mathematics

Member of
Committee

Date of thesis/dissertation defense: September 3, 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Assidi Maya Adel
Last First Middle

Master's Thesis Master's Project Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.



September 10, 2021

Signature

Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgments

I would like to express my gratitude to my advisor Prof. Nabil Nassif for encouraging and helping me. Thanks for his motivation and patience.

My deep appreciation goes equally to my co-advisor, Prof. Sophie Moufawad for being there every time I needed your help and for guiding me through the testing part of this thesis and for being a member of my committee.

I also thank my committee members: Professor Ghassan Antar and Professor Ahmad Sabra for their encouragement.

Finally, I thank my family and friend for supporting me, throughout my life and being part of my success.

An Abstract of the Thesis of

Maya Adel Assidi for Master of Science
Major: Mathematics

Title: Newton Type Methods for Solving Finite-Element Space, Euler-Implicit Time Discretization of the Hasegawa-Mima Equation

Hasegawa-Mima was derived by Akira Hasegawa and Kunioki Mima during late 70s. When normalized, it can be put as the following PDE that is third order in space and first order in time:

$$-\Delta u_t + u_t = \{u, \Delta u\} + \{p, u\}$$

In a recent work [9], this equation was reformulated as a coupled system and put in variational form. Also, a continuous Time Integral Formulation was derived over any time interval. Consequently, a finite element space semi-discretization followed by Euler-Implicit time discretization was obtained and extensively studied in [9].

In this thesis, we focus on solving this fully discrete system using Newton's type methods: Full Newton, Quasi Newton (Chord Variant), and a newly introduced Modified Newton. The tests are based on several initial data and the implementation uses Free-FEM ++ software [6] to obtain a conformal meshing and solve the linearized system.

Test results indicate alternative efficient and robust algorithms to the one used in [9].

Contents

Acknowledgments	v
Abstract	vi
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Hasegawa-Mima (HM) Model for magnetized Plasma Physics .	1
1.1.1 Introduction on Energy and Plasma Physics	1
1.1.2 Hasegawa Mima Simplified Mathematical Model	2
1.2 Literature Review	6
1.3 Outline of the Thesis	7
2 Discretizing the Hasegawa-Mima model as a coupled system of Partial Differential Equations	8
2.1 Definition of Spaces	8
2.1.1 Periodic Sobolev spaces for $p = 2$	9
2.2 Variational formulation of the Hasegawa-Mima Equation as a Coupled System of Partial Differential Equations	10
2.3 Variational Formulation of (HM) equation as a coupled system	11
2.3.1 Variational formulation of Hyperbolic Part of the coupled system problem	11
2.3.2 Variational Formulation of the Poisson Elliptic equation	14
2.3.3 Semi-Variational Formulation	16
2.3.4 Time Integral Formulation	16
2.4 Discretization	17
2.4.1 Finite Element Space Semi-Discretization	17
2.4.2 Full Discretization	22
3 Solving Fully Discrete Hasegawa-Mima System	25
3.1 Formulation of Newton's method	25

3.2	Variants of Newton's Method	29
4	Testing	34
4.1	Newton-Type Methods	34
4.1.1	Test 1	34
4.1.2	Test 2	37
4.1.3	Test 3	40
4.1.4	Comparison and Analysis of results	43
4.2	Semi-linear versus Modified Newton's Method	45
4.2.1	Test 1	45
4.2.2	Test 2	47
4.2.3	Test 3	50
4.2.4	Comparison and Analysis of results	52
5	Concluding Remarks	53
	Bibliography	55

List of Figures

2.1	The boundary of the domain Ω	12
2.2	A conformal finite element meshing of the rectangle ($n \times n$)	18
2.3	Right Triangle in Ω	19
2.4	Domain Ω	21
4.1	Time evolution of solution u using full newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	35
4.2	Time evolution of solution u using chord method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	35
4.3	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	36
4.4	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, \pi] \times [0, \pi]$	36
4.5	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$	37
4.6	Time evolution of solution u using full newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	38
4.7	Time evolution of solution u using chord method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	38
4.8	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$	39
4.9	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, \pi] \times [0, \pi]$	39

4.10	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$	40
4.11	Time evolution of solution u using full newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$	41
4.12	Time evolution of solution u using chord method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$	41
4.13	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$	42
4.14	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, 20] \times [0, 20]$	42
4.15	Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$	43
4.16	Time evolution of solution u for test 1 using Modified Newton method for $T = 200$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$	46
4.17	Time evolution of solution u for test 1 using semi-linear for $T = 200$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$	47
4.18	Time evolution of solution u for test 2 using Modified Newton method for $T = 200$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$	48
4.19	Time evolution of solution u for test 2 using semi-linear for $T = 200$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$	49
4.20	Time evolution of solution u for test 3 using Modified Newton method for $T = 200$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$	50
4.21	Time evolution of solution u for test 3 using semi-linear for $T = 200$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$	51

List of Tables

4.1	Runtimes of the three methods for three the tests number of intervals in the x and y directions, $n = 16$ and $T = 50$	44
4.2	Number of iterations k per time step	44
4.3	Runtimes of Modified Newton's methods for the three tests with different number of intervals in the x and y directions n for $T = 50$	44
4.4	Variation of t_{max} for semi-linear and Modified Newton's method in three tests, for $n = 64$ and $T = 200$	52
4.5	Runtimes of the semi-linear and Modified Newton's method for test 3 with number of intervals in x and y directions, $n = 64$ and $T = 200$	52

Chapter 1

Introduction

1.1 Hasegawa-Mima (HM) Model for magnetized Plasma Physics

1.1.1 Introduction on Energy and Plasma Physics

Despite the growing interest and policies in saving energy, global energy demand is rising due to population growth, economic development, technology developments, and industrial activity in both developing and developed country. Such condition has come to a cost especially to irreplaceable fossil fuel energy resources that will be insufficient for humanity energy production for more than 150 years. Consequently, finding alternative energies became a necessity. Thermonuclear fusion becomes an interesting field of study as its primary fuel sources are deuterium and tritium which are isotopes of hydrogen, the most abundant element in the universe. This is why it has the potential to provide almost limitless energy for mankind. In addition, nuclear power releases into environment less radiation than other major energy resources. Moreover, nuclear power plants operate at much higher capacity factors than renewable resources or fossil fuels due to its potentials which made thermonuclear energy better source than intermittent energy sources: the sun doesn't shine always, nor the wind blows always, nor the water fall permanently. Fusion energy research is widespread in the US and Europe and currently more popular in Asia (e.g. China, India and Korea) and South America (e.g. Mexico and Brazil). Such field is very significant and has a high strategic position as Thousands of engineers and scientists have contributed to the design of ITER, the world biggest toroidal fusion device, since the idea for an international joint experiment in fusion was first launched in 1985. The ITER Members—China, the European Union, India, Japan, Korea, Russia and the United States—are now engaged in a 35-year collaboration to build and operate the ITER experimental device, and together bring technology where

fusion reactor can be designed and tested.

Nuclear fusion is a process in which two light atomic nuclei are combined to form a heavier one while releasing massive amount of energy. Large scale fusion processes must occur in a plasma state matter.

Plasma is the fourth state of matter state and it is the most common one in universe as a whole. Like gas, it has no fixed shape or volume state of matter. In solids, the atoms are strongly related, so if we heat it up we get a liquid, and if we heat liquids we get gases, and if we heat gases we get what is called a Plasma. It is an ionized gas in which a certain proportion of electrons are free rather than being bounded to an atom or molecule. It is a region where charges of ions and electrons are balanced and free. Most plasmas are quasi neutral i.e.: negatively charged electrons are almost completely neutralized by positively charged ions. That is $n_e = n_i$, where n , is the plasma density usually refers to the electron or ion density (n_e or n_i), that is, the number of free particles per unit volume. It is electrically conductive due to the presence of free charged particles. A plasma in which the magnetic field is strong enough to influence the motion of the charged particles is said to be a magnetized plasma.

Ionization is necessary for existence of plasma. For this purpose, the input energy is greater than the bonding energy. Plasma ionization is the proportion of atoms which have lost electrons and is determined by the electron temperature T_e relative to the ionization energy.

We define the current density \vec{j} to be

$$\vec{j} = ne(\vec{u}_i - \vec{u}_e),$$

where n is the density, e is the charge and u_i , u_e respectively the ion and electron velocities.

Note that if the velocity of the electrons is equal to that of the ions, there is no current.

Moreover we have Ohm's law:

$$\vec{j} = \sigma \vec{E},$$

where \vec{E} is electric field and σ is the conductivity.

Therefore, in a plasma, charge separation between ions and electrons gives rise to electric fields, and different charged-particle gives rise to currents and magnetic fields.

1.1.2 Hasegawa Mima Simplified Mathematical Model

In this section, following Hariri [2], we present the derivation of a reduced model for plasma turbulence called the Hasegawa-Mima equation which is governed by:

1. The particle balance equation, known also as **the continuity equation**:

$$\partial_t n + \nabla \cdot (n \vec{u}) = 0, \quad (1.1.1)$$

where n is the density, t is time and \vec{u} is the fluid velocity. In case of incompressible flow, where the density n is constant, such equation can be simplified to $\nabla \cdot \vec{u} = 0$, which means that the divergence of velocity field is zero everywhere.

The continuity equation (1.1.1) is equivalent to

$$\partial_t n + n \nabla \cdot \vec{u} + \vec{u} \cdot \nabla n = 0 \quad (1.1.2)$$

Dividing Eq. (1.1.2) by n yields

$$\frac{\partial_t n}{n} + \nabla \cdot \vec{u} + \frac{\vec{u} \cdot \nabla n}{n} = 0 \quad (1.1.3)$$

Using

$$\begin{aligned} \frac{\partial_t n}{n} &= \partial_t \ln(n) \\ \frac{\nabla n}{n} &= \nabla \ln(n), \end{aligned}$$

Eq. (1.1.3) implies

$$\partial_t \ln(n) + \nabla \cdot \vec{u} + \vec{u} \cdot \nabla \ln(n) = 0 \quad (1.1.4)$$

According to the definition of convective derivative

$$d_t = \partial_t + \vec{u} \cdot \nabla, \quad (1.1.5)$$

we get

$$d_t \ln(n) + \nabla \cdot \vec{u} = 0. \quad (1.1.6)$$

Using the adiabatic relation:

$$n = n_0 \exp\left(\frac{e\phi}{T}\right), \quad (1.1.7)$$

where $n_0 = n_0(x, y)$ is the particle density depending on the x -direction and/or on the y -direction, T is the temperature and e is the electron charge. We assume also that both ions and electrons are characterized by the same temperature T . Finally, $\phi = \phi(x, y, t)$ represents the electrostatic potential fluctuations. Hence, one has:

$$\ln(n) = \ln(n_0) + \frac{e\phi}{T} \quad (1.1.8)$$

Using (1.1.8), Eq. (1.1.6) is equivalent to

$$d_t \ln(n_0) + d_t \left(\frac{e\phi}{T} \right) + \nabla \cdot \vec{u} = 0 \quad (1.1.9)$$

$$\iff \partial_t \ln(n_0) + \vec{u} \cdot \nabla \ln(n_0) + d_t \left(\frac{e\phi}{T} \right) + \nabla \cdot \vec{u} = 0. \quad (1.1.10)$$

But $\partial_t \ln(n_0) = 0$ since n_0 is independent of time t , therefore:

$$\vec{u} \cdot \nabla \ln(n_0) + d_t \left(\frac{e\phi}{T} \right) + \nabla \cdot \vec{u} = 0. \quad (1.1.11)$$

We write $\vec{u} = \vec{u}_E + \vec{u}_p$ where u_E is the electric field velocity and u_p is the potential velocity. Since $|u_p| \ll |u_E|$, therefore $\vec{u} \cdot \nabla \ln(n_0) \approx \vec{u}_E \cdot \nabla \ln(n_0)$. Moreover, since $\nabla \cdot \vec{u}_E = 0$ (i.e., \vec{u}_E is divergence-free), then $\nabla \cdot \vec{u} = \nabla \cdot \vec{u}_p$.

Thus, the continuity equation reduces to:

$$\nabla \cdot \vec{u}_p + \vec{u}_E \cdot \nabla \ln(n_0) + d_t \left(\frac{e\phi}{T} \right) = 0 \quad (1.1.12)$$

2. The second governing principle is the **Momentum balanced equation**:

$$mnd_t \vec{u} = -\nabla p + nq(\vec{E} + \vec{u} \times \vec{B}),$$

where m is the mass, n is the particle density, t is time, \vec{u} is the velocity, p is the pressure, q is the charge of the particle, \vec{E} is the electric field, \vec{B} is the magnetic field. By projecting it on the direction parallel to the magnetic field \vec{B} , we obtain the adiabatic relationship (1.1.7).

Given the adiabatic assumption, the electrons follow immediately the ions in the perpendicular direction to \vec{B} . Thus, we obtain the HM equation for the potential

$$d_t \left[\left(\frac{e\phi}{T} \right) - \frac{1}{w_{ci} B} \nabla_{\perp}^2 \phi \right] + \vec{u}_E \cdot \nabla \ln \left(\frac{n_0}{w_{ci}} \right) = 0 \quad (1.1.13)$$

where, ∇_{\perp} is the projection of $\vec{\nabla}$ on the direction perpendicular to the magnetic field \vec{B} , $w_{ci} = \frac{eB_0}{m_i}$ is the ion cyclotron frequency that depends only on the initial magnetic field B_0 .

Taking into consideration the adiabatic assumption, and since we are considering the flow in the perpendicular direction to the magnetic field, we replace ∇_{\perp} by ∇ . Then, eq. (1.1.13) is normalized to obtain the Hasegawa-Mima equation which can be written as follows:

$$\frac{\partial}{\partial t} (\vec{\nabla}^2 \phi - \phi) - [(\vec{\nabla} \phi \times \hat{z}) \cdot \nabla] [\nabla^2 \phi - \ln \left(\frac{n_0}{w_{ci}} \right)] = 0 \quad (1.1.14)$$

$$\iff \frac{\partial}{\partial t}(\Delta\phi - \phi) - [(\nabla\phi \times \hat{z}) \cdot \vec{\nabla}][\Delta\phi - \ln(\frac{n_0}{w_{ci}})] = 0 \quad (1.1.15)$$

where $\hat{z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$.

The Hasegawa-Mima equation can be expressed in a different form using the Poisson brackets. Poisson brackets are defined as follows:

$$\{r, s\} = \partial_x r \partial_y s - \partial_y r \partial_x s.$$

We have:

$$\vec{\nabla}\phi \times \hat{z} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \partial_x\phi & \partial_y\phi & 0 \\ 0 & 0 & 1 \end{vmatrix} = \begin{pmatrix} \partial_y\phi \\ -\partial_x\phi \\ 0 \end{pmatrix}$$

$$[(\nabla\phi \times \hat{z}) \cdot \nabla] = \begin{pmatrix} \partial_y\phi \\ -\partial_x\phi \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} = \partial_y\phi\partial_x - \partial_x\phi\partial_y$$

$$\begin{aligned} [(\nabla\phi \times \hat{z}) \cdot \nabla][\Delta\phi - \ln(\frac{n_0}{w_{ci}})] &= [\partial_y\phi\partial_x - \partial_x\phi\partial_y][\Delta\phi - \ln(\frac{n_0}{w_{ci}})] \\ &= \partial_y\phi\partial_x\Delta\phi - \partial_x\phi\partial_y\Delta\phi - \partial_y\phi\partial_x \ln(\frac{n_0}{w_{ci}}) + \partial_x\phi\partial_y \ln(\frac{n_0}{w_{ci}}) \\ &= \{\Delta\phi, \phi\} + \{\phi, \ln(\frac{n_0}{w_{ci}})\} \end{aligned}$$

Then, (1.1.15) can be expressed using Poisson bracket as follows:

$$\partial_t(\Delta\phi - \phi) - \{\Delta\phi, \phi\} - \{\phi, \ln(\frac{n_0}{w_{ci}})\} = 0 \quad (1.1.16)$$

$$\iff -\partial_t(\Delta\phi - \phi) = -\{\Delta\phi, \phi\} - \{\phi, \ln(\frac{n_0}{w_{ci}})\} \quad (1.1.17)$$

$$\iff -\Delta\phi_t + \phi_t = \{\phi, \Delta\phi\} + \{\ln(\frac{n_0}{w_{ci}}), \phi\} \quad (1.1.18)$$

(since $\{r, s\} = -\{s, r\}$).

From hereon and for the rest of the thesis, we will be replacing ϕ by u , thus considering the PDE:

$$-\Delta u_t + u_t = \{u, \Delta u\} + \{p, u\} \quad (1.1.19)$$

where:

- $\{u, v\} = u_x v_y - u_y v_x$.
- $u(x, y, t)$ describes electrostatic potential.
- Δ is the Laplacian operator i.e: $\Delta u = u_{xx} + u_{yy}$
- $p = \ln\left(\frac{n_0}{w_{ci}}\right)$ is a function depending on the background particle density n_0 and the ion cyclotron frequency w_{ci} that depends only on the initial magnetic field

In this context, $p = 0$ refers to homogeneous plasma, and $p \neq 0$ refers to non-homogeneous plasma.

In this thesis, we deal with the numerical solution to Hasegawa-Mima equation on a rectangular domain with the solution u , satisfying periodic boundary conditions (PBCs). For that purpose, we consider $\Omega = (0, L) \times (0, L) \in \mathbb{R}^2$ and use the frame of periodic Sobolev spaces which are closed subspace of $H^m(\Omega)$, and therefore itself a Hilbert space.

Let $\Omega = (0, L) \times (0, L)$, note that $\partial\Omega$ is boundary of Ω

Given an initial data $u_0: \bar{\Omega} \rightarrow \mathbb{R}$, we seek $u: \bar{\Omega} \times [0, T] \rightarrow \mathbb{R}$

$$\begin{cases} (\Delta - I)u_t + \{u, \Delta u\} + p_x u_y - p_y u_x = 0 & \text{on } \Omega \times (0, T] \\ u(x, y, 0) = u_0(x, y) & \text{on } \Omega \\ u, \nabla u \text{ satisfies PBCs} & \text{on } \partial\Omega \times (0, T] \end{cases} \quad (1.1.20)$$

without loss of generality, it is assumed that the background particle density n_0 is a function of x only, such that $p_x = k$ is a constant and $p_y = 0$, i.e. $n_0 = e^{Ax+B}$ for $A, B \in \mathbb{R}$.

- Periodic boundary conditions for u and Δu :

$$\begin{cases} u(0, y) = u(L, y) & \forall y \in (0, L) \\ u(x, 0) = u(x, L) & \forall x \in (0, L) \\ u_x(0, y) = u_x(L, y) & \forall y \in (0, L) \\ u_y(x, 0) = u_y(x, L) & \forall x \in (0, L) \end{cases}$$

1.2 Literature Review

Solving the full Navier-Stokes equations is extremely complicated, instead simplified models to describe turbulence were first developed by Hasegawa-Mima (HM) and Hasegawa Wakatani [5]. The physics basics of the Hasegawa-Mima equation were studied in [12]. It is a simplified two-dimensional non-linear equation with one variable which is the electrostatic potential. It was

found that the HM equation has an exact solution of the form of a stationary dipole vortex traveling in the direction perpendicular to the density gradient [10, 11]. Several authors (Hasegawa and Mima [3, 4], Terry and Horton [7, 8, 13, 14] and Waltz [15]) have studied two-dimensional drift wave turbulence. Harriri in [2] represented in detail the physics basics behind the simulate plasma turbulence using the latter model equation. A computer model was designed for solving the two-dimensional Hasegawa-Mima equation based on a finite difference (FD) approach with the integration in time being carried out with a Euler explicit scheme that constraints the time-step size which limit the size of the time interval. Hence, such method is not well suited for periodic boundary conditions, has a major difficulty in discretizing the Poisson-bracket term. Thus, it is not applicable for computations on long time intervals. Maalouf [1] transformed the HM equation to a coupled system and then performed a new numerical simulation with the adequate boundary conditions and initial conditions. In addition, the modon steadiness solution for the nonlinear Hasegawa- Mima equation was studied in order to test the Periodic Boundary Conditions using finite element method. Then, N-Nassif, S- Moufawad and H-Karakazian in their recent work [9] used a finite element space-domain approach to semi-discretize the coupled variational Hasegawa-Mima model. Thus they obtained global existence of solutions in H^2 on any time interval $[0; T]; \forall T$. They carried tests on semi-linear version of the implicit nonlinear full-discrete system for several initial data. Moreover, they derived an implicit full-discretized system and proved the existence and uniqueness of solution using fixed point approach. However, there is no results for solving such full discretized system.

1.3 Outline of the Thesis

In addition to the introduction, the thesis consists of 4 chapters including the concluding remarks. In Chapter 2, we use a finite element space-domain approach to full-discrete the coupled variational Hasegawa- Mima model and summarize the theoretical results recently obtained by N-Nassif, S-Moufawad, H- Karakazian in their recent work [9]. In Chapter 3, we discuss Newton type methods (Full Newton, Quasi Newton, Modified Newton) to solve the fully discretize system of Hasegawa-Mima model and present for each variant of Newton's Method the corresponding algorithm using Free-FEM++ software. Consequently, in Chapter 4 , we give the results of our consequent numerical simulations. Then, we analyze and compare the numerical solutions of 3 Newton Type methods with that of the solution obtained in [9] using the semi-linearized discretization. Tests were conducted on various initial conditions. Finally, concluding remarks are given at the end of the thesis.

Chapter 2

Discretizing the Hasegawa-Mima model as a coupled system of Partial Differential Equations

In this chapter, we summarize the theoretical results recently obtained by N-Nassif , S-Moufawad, H- Karakazian in their recent work [9]

2.1 Definition of Spaces

In this section, we are going to review the definition of some useful spaces reaching the construction of Periodic Sobolev Spaces

- The $C_0^\infty(\Omega)$ Space = $\{f \in C^\infty(\Omega) | f \text{ is compactly supported in } \Omega\}$.
- The L^p Spaces = $\{f \text{ is measurable } | \int_\Omega |f|^p dx < \infty\}$ where $p \in [1, \infty)$ }.
In particular, $L^2(\Omega) = \{f \text{ is measurable } | \int_\Omega |f|^2 dx < \infty\}$
- A function $f \in L^2(\Omega)$ is said to have a weak derivative $D_w f \in L^2(\Omega)$ if and only if $\forall \psi \in C_0^\infty(\Omega), \langle f, D_w \psi \rangle = - \langle D_w f, \psi \rangle$.
- $D^\alpha u$ in space $\Omega \subset \mathbb{R}^2$ where $\alpha = (\alpha_1, \alpha_2)$ is defined as follows:

$$D^\alpha u = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} u$$

For example:

$$\alpha = (2, 0), |\alpha| = 2, \text{ so } D^{(2,0)} u = \frac{\partial^2 u}{\partial x_1^2}$$
$$\alpha = (1, 1), |\alpha| = 2, \text{ so } D^{(1,1)} u = \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} u$$

- Sobolev spaces $W^{k,p}(\Omega) = \{u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega) \forall |\alpha| \leq k\}$, $p \in [1, \infty]$, $k \in \mathbb{N}$

The natural number k is called the order of Sobolev space $W^{k,p}(\Omega)$

$W^{k,p}(\Omega)$ is Banach space with respect norm $\|\cdot\|_{W^{k,p}(\Omega)}$:

$$\|u\|_{W^{k,p}(\Omega)} = \begin{cases} (\sum_{|\alpha| \leq k} \|D^\alpha u\|_{L^p(\Omega)}^p)^{\frac{1}{p}} & p \in [1, \infty) \\ \max_{|\alpha| \leq k} \|D^\alpha u\|_{L^\infty(\Omega)} & p = \infty \end{cases}$$

In special case of $p = 2$, we write $W^{k,2}(\Omega) := H^k(\Omega)$ in which $\|\cdot\|_{W^{k,2}(\Omega)} = \|\cdot\|_{H^k(\Omega)}$ is Hilbertian norm associated to the inner product

$$(f, g)_{H^k(\Omega)} = \sum_{|\alpha| \leq k} \int_{\Omega} D^\alpha f \overline{D^\alpha g} \, dx.$$

2.1.1 Periodic Sobolev spaces for $p = 2$

Starting with the Sobolev spaces of order 0, 1, 2 and 3:

$$\begin{cases} H^0(\Omega) = L^2(\Omega) \\ H^1(\Omega) = \{f \in L^2(\Omega) \mid D^\alpha f \in L^2(\Omega), |\alpha| = 1\} \\ H^2(\Omega) = \{f \in L^2(\Omega) \mid D^\alpha f \in L^2(\Omega), |\alpha| \in \{1, 2\}\} \\ H^3(\Omega) = \{f \in L^2(\Omega) \mid D^\alpha f \in L^2(\Omega), |\alpha| \in \{1, 2, 3\}\} \end{cases}$$

We introduce Periodic Boundary Conditions.

A function v satisfies the periodic boundary conditions PBC^k of order k if and only if for $k \geq 0$

$$(\text{PBC}^k) \begin{cases} (\text{PBC}_x^k) : \partial_x^k v(0, y) = \partial_x^k v(L, y) \quad \forall y \in (0, L) \\ (\text{PBC}_y^k) : \partial_y^k v(x, 0) = \partial_y^k v(x, L) \quad \forall x \in (0, L) \end{cases}$$

where ∂_x^k and ∂_y^k denote the differential operators $\frac{\partial^k}{\partial x^k}$ and $\frac{\partial^k}{\partial y^k}$ respectively.

Consequently, we define the Periodic Sobolev space of order 0, 1, 2 and 3 as:

$$\begin{cases} H_p^0(\Omega) = L^2(\Omega) \quad \text{and} \quad H_p^\infty(\Omega) := \bigcap_{m \geq 1} \{H_p^m\}, \\ H_p^1(\Omega) = \left\{ u \in H^1(\Omega) \mid \begin{array}{l} u(x, 0) = u(x, L), \quad x \in (0, L) \text{ a.e.} \\ u(0, y) = u(L, y), \quad y \in (0, L) \text{ a.e.} \end{array} \right\} \\ H_p^2(\Omega) = \{u \in H^2(\Omega) \mid u, u_x, u_y \in H_p^1(\Omega)\} \\ H_p^3(\Omega) = \{u \in H^3(\Omega) \mid u, u_x, u_y, u_{xx}, u_{yy}, u_{xy}, u_{yx} \in H_p^1(\Omega)\} \end{cases}$$

2.2 Variational formulation of the Hasegawa-Mima Equation as a Coupled System of Partial Differential Equations

In this section, we formulate from (1.1.20) a coupled system of linear Elliptic-Hyperbolic coupled system problem in order to overcome the theoretical and computational issues of them non-linearity of the Poisson bracket $\{u, \Delta u\} = u_x(\Delta u_y) - u_y(\Delta u_x)$

The Formulation of Elliptic-Hyperbolic coupled system problem:

Given a time $T > 0$, we consider the Hasegawa-Mima problem on a square domain $\Omega = (0, L) \times (0, L)$ with boundary $\partial\Omega$.

Given an initial data $u_0: \overline{\Omega} \rightarrow \mathbb{R}$, we seek $u: \overline{\Omega} \times [0, T] \rightarrow \mathbb{R}$

$$\begin{cases} (\Delta - I)u_t + \{u, \Delta u\} + ku_y = 0 & \text{on } \Omega \times (0, T] \\ u(x, y, 0) = u_0(x, y) & \text{on } \overline{\Omega} \\ u, \nabla u \text{ satisfies PBCs} & \text{on } \partial\Omega \times (0, T] \end{cases} \quad (2.2.1)$$

where $\{u, v\} = u_x v_y - u_y v_x$ is the Poisson bracket which is bilinear.

Since handling the non-linearity of Poisson bracket $\{u, \nabla u\}$ is both theoretically and computationally expensive, we formulate (HM) as a coupled system of linear equations as follows introduce the variable w such that

$$w = -\Delta u + u \quad (2.2.2)$$

Then, $\frac{\partial w}{\partial t} = -\Delta u_t + u_t = -(\Delta - I)u_t = \{u, \Delta u\} + ku_y = \{u, u - w\} + ku_y = \{u, u\} + \{u, -w\} + ku_y = \{w, u\} + ku_y = w_x u_y - w_y u_x + ku_y$
(since $\{u, u\} = u_x u_y - u_y u_x = 0$ and $\{u, -w\} = -u_x w_y + u_y w_x = \{w, u\}$)

So, $\frac{\partial w}{\partial t} - w_x u_y + w_y u_x = ku_y$.

Now define the divergence free vector field $\vec{V}(u) = \begin{pmatrix} -u_y \\ u_x \end{pmatrix}$, the following equation is obtained :

$$\frac{\partial w}{\partial t} + \vec{V}(u) \cdot \nabla w = ku_y \quad (2.2.3)$$

Now, we formulate using (2.2.2) and (2.2.3) the (HM) problem as the following Elliptic-Hyperbolic coupled system problem, where one seeks

$(u, w) : \bar{\Omega} \times [0, T] \rightarrow \mathbb{R}^2$ such that:

$$\begin{cases} u - \Delta u = w & \text{on } \Omega \times (0, T] \\ \frac{\partial w}{\partial t} + \vec{V}(u) \cdot \nabla w = k u_y & \text{on } \Omega \times (0, T] \\ u, \nabla u \text{ and } w \text{ satisfies PBCs} & \text{on } \partial\Omega \times (0, T] \\ u(x, y, 0) = u_0(x, y) \text{ and } w(x, y, 0) = w_0 & \text{on } \bar{\Omega} \end{cases} \quad (2.2.4)$$

2.3 Variational Formulation of (HM) equation as a coupled system

In this section, we intend to formulate a strong semi-variational form from Elliptic-Hyperbolic coupled system problem (2.2.4).

2.3.1 Variational formulation of Hyperbolic Part of the coupled system problem

We start with Variational formulation of the two dimensional Hyperbolic part satisfying periodic boundary conditions

$$\begin{cases} w_t + \vec{V}(u) \cdot \nabla w = k u_y \\ w(x, y, 0) = w_0(x, y) = -\Delta u_0 + u_0 \end{cases} \quad (2.3.1)$$

where the initial conditions $u_0 \in H^2(\Omega) \cap H_p^1(\Omega)$ and $w_0 \in H_p^1(\Omega)$.

Given $u \in C([0, T]; H^2(\Omega) \cap H_p^1(\Omega)) \cap C^1((0, T); L^2(\Omega))$, where u_x and u_y satisfy PBCs, one seeks of $w \in C([0, T], H_p^1(\Omega)) \cap \{C^1((0, T), L^2(\Omega))\}$

Choose the test function $v \in H_p^1(\Omega)$, then multiply the first equation of (2.3.1) by a test function v and integrate with respect to Ω

$$\int_{\Omega} w_t v \, dx \, dy + \int_{\Omega} (\vec{V}(u) \cdot \nabla w) v \, dx \, dy = \int_{\Omega} k u_y v \, dx \, dy \quad (2.3.2)$$

$$\langle w_t, v \rangle_{L^2(\Omega)} + \langle \vec{V}(u) \cdot \nabla w, v \rangle_{L^2(\Omega)} = \langle k u_y, v \rangle_{L^2(\Omega)} \quad (2.3.3)$$

According to Green's theorem

$$\int_{\Omega} (\vec{V}(u) \cdot \nabla w) v \, dx \, dy = \int_{\Gamma} w \vec{v} \cdot (v \vec{V}(u)) \, ds - \int_{\Omega} w \vec{V}(u) \cdot \nabla v \, ds \quad (2.3.4)$$

where $\vec{V}(u) = \begin{pmatrix} -u_y \\ u_x \end{pmatrix}$ and \vec{v} is the outer normal on $\Gamma = \partial\Omega$.

We prove then $\int_{\Gamma} \vec{v} \vec{V}(u) \cdot w v \, ds = 0$ where $\Gamma = \partial\Omega$ is the boundary of the domain Ω represented by the the square ABCD (2.1) where [AB] and [CD] opposite horizontal sides, [AD] and [CB] opposite vertical sides.

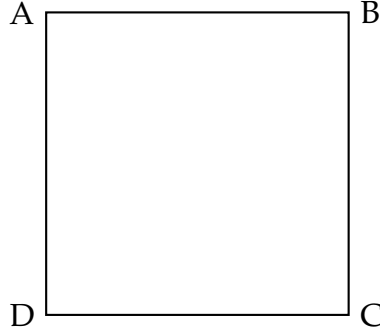


Figure 2.1: The boundary of the domain Ω

Define the four maps $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ such that :

$$\begin{aligned} \Gamma_1: [0, L] &\rightarrow [AB] \\ s &\mapsto (s, 0) \end{aligned}$$

$$\begin{aligned} \Gamma_2: [0, L] &\rightarrow [BC] \\ s &\mapsto (L, s) \end{aligned}$$

$$\begin{aligned} \Gamma_3: [0, L] &\rightarrow [CD] \\ s &\mapsto (L - s, L) \end{aligned}$$

$$\begin{aligned} \Gamma_4: [0, L] &\rightarrow [DA] \\ s &\mapsto (0, L - s) \end{aligned}$$

$$\begin{aligned} \int_{\Gamma} \vec{v} \cdot \vec{V}(u) w v ds &= \int_{\Gamma_1} w v (0 \quad -1) \begin{pmatrix} -u_y \\ u_x \end{pmatrix} ds + \int_{\Gamma_2} w v (1 \quad 0) \begin{pmatrix} -u_y \\ u_x \end{pmatrix} ds \\ &+ \int_{\Gamma_3} w v (0 \quad 1) \begin{pmatrix} -u_y \\ u_x \end{pmatrix} ds + \int_{\Gamma_4} w v (-1 \quad 0) \cdot \begin{pmatrix} -u_y \\ u_x \end{pmatrix} ds \\ &= - \int_{\Gamma_1} w v u_x ds - \int_{\Gamma_2} w v u_y ds + \int_{\Gamma_3} w v u_x ds + \int_{\Gamma_4} w v u_y ds \end{aligned}$$

Note that:

$$\int_{\Gamma_1} w v u_x ds = \int_0^L w(s, 0) v(s, 0) u_x(s, 0) ds \quad (2.3.5)$$

$$\int_{\Gamma_3} w v u_x ds = \int_0^L w(L - s, L) v(L - s, L) u_x(L - s, L) ds \quad (2.3.6)$$

apply change of variable $r = L - s$ on (2.3.6), and since u_x , w and v satisfy PBCs,

$$\begin{aligned} \int_{\Gamma_3} wvu_x ds &= \int_0^L w(L-s, L)v(L-s, L)u_x(L-s, L) ds = - \int_L^0 w(r, L)v(r, L)u_x(r, L) dr \\ &= \int_0^L w(r, L)v(r, L)u_x(r, L) dr = \int_0^L w(s, L)v(s, L)u_x(s, L) ds = \int_{\Gamma_1} wvu_x ds \end{aligned}$$

so,

$$\int_{\Gamma_3} wvu_x ds - \int_{\Gamma_1} wvu_x ds = 0 \quad (2.3.7)$$

In the same manner,

$$\int_{\Gamma_2} wvu_y ds = \int_0^L w(L, s)v(L, s)u_y(L, s) ds \quad (2.3.8)$$

$$\int_{\Gamma_4} wvu_y ds = \int_0^L w(0, L-s)v(0, L-s)u_y(0, L-s) ds \quad (2.3.9)$$

apply change of variable $r = L - s$ on (2.3.9), and since u_y , w and v satisfy PBCs,

$$\begin{aligned} \int_{\Gamma_4} wvu_y ds &= \int_0^L w(0, L-s)v(0, L-s)u_y(0, L-s) ds = - \int_L^0 w(0, r)v(0, r)u_y(0, r) dr \\ &= \int_0^L w(0, r)v(0, r)u_y(0, r) dr = \int_0^L w(0, s)v(0, s)u_y(0, s) ds = \int_{\Gamma_2} wvu_y ds \end{aligned}$$

so,

$$\int_{\Gamma_4} wvu_y ds - \int_{\Gamma_2} wvu_y ds = 0 \quad (2.3.10)$$

By (2.3.7) and (2.3.10) ,

$$\int_{\Gamma} \vec{v} \cdot \vec{V}(u) wv ds = - \int_{\Gamma_1} wvu_x ds - \int_{\Gamma_2} wvu_y ds + \int_{\Gamma_3} wvu_x ds + \int_{\Gamma_4} wvu_y ds = 0$$

so, (2.3.4) becomes

$$\int_{\Omega} (\vec{V}(u) \cdot \nabla w) v dx dy = - \int_{\Omega} (\vec{V}(u) \cdot \nabla v) w dx dy$$

Therefore, (2.3.2) implies

$$\int_{\Omega} w_t v - \int_{\Omega} (\vec{V}(u) \cdot \nabla v) w dx dy = \int_{\Omega} vku_y dx dy$$

Thus the following variational formulation is obtained:

$$\underbrace{\langle w_t, v \rangle_{L^2(\Omega)} - \langle \vec{V}(u) \nabla v, w \rangle_{L^2(\Omega)}}_{B(w, v)} = \underbrace{\langle ku_y, v \rangle_{L^2(\Omega)}}_{G(v)} \quad (2.3.11)$$

2.3.2 Variational Formulation of the Poisson Elliptic equation

The Variational formulations of the Poisson Elliptic equation with periodic boundary conditions is carried out in the same way as for the hyperbolic equation. Consider the elliptic part of the coupled system problem (2.2.4)

$$\begin{cases} -\Delta u + u = w \\ u(0) = u_0 \\ u, u_x \text{ and } u_y \text{ satisfy PBCs} \end{cases} \quad (2.3.12)$$

Multiply the first equation of (2.3.12) by the test function $v \in H_p^1(\Omega)$ and integrate both sides with respect to Ω .

$$-\int_{\Omega} (\text{div}(\nabla u))v \, dx \, dy + \int_{\Omega} uv \, dx = \int_{\Omega} wv \, dx \, dy \quad (2.3.13)$$

According to Green's theorem

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx \, dy - \int_{\Gamma} \vec{v} \cdot \nabla u \, dx \, dy = \int_{\Omega} -\text{div}(\nabla u)v \, dx \, dy. \quad (2.3.14)$$

Similarly, we prove that $\int_{\Gamma} \vec{v} \cdot \nabla u \, dx \, dy = 0$ where $\Gamma = \partial\Omega$ as shown in 2.1 Define the four maps $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ such that:

$$\begin{aligned} \Gamma_1: [0, L] &\rightarrow [AB] \\ s &\mapsto (s, 0) \end{aligned}$$

$$\begin{aligned} \Gamma_2: [0, L] &\rightarrow [BC] \\ s &\mapsto (L, s) \end{aligned}$$

$$\begin{aligned} \Gamma_3: [0, L] &\rightarrow [CD] \\ s &\mapsto (L - s, L) \end{aligned}$$

$$\begin{aligned} \Gamma_4: [0, L] &\rightarrow [DA] \\ s &\mapsto (0, L - s) \end{aligned}$$

$$\begin{aligned} \int_{\Gamma} \vec{v} \cdot \vec{V}(u)v \, ds &= \int_{\Gamma_1} v(0 \quad -1) \begin{pmatrix} u_x \\ u_y \end{pmatrix} ds + \int_{\Gamma_2} v(1 \quad 0) \begin{pmatrix} u_x \\ u_y \end{pmatrix} ds \\ &+ \int_{\Gamma_3} v(0 \quad 1) \begin{pmatrix} u_x \\ u_y \end{pmatrix} ds + \int_{\Gamma_4} v(-1 \quad 0) \cdot \begin{pmatrix} u_x \\ u_y \end{pmatrix} ds \\ &= - \int_{\Gamma_1} vu_y \, ds + \int_{\Gamma_2} vu_x \, ds + \int_{\Gamma_3} vu_y \, ds - \int_{\Gamma_4} vu_x \, ds \end{aligned}$$

Note that,

$$\int_{\Gamma_1} v u_y ds = \int_0^L v(s, 0) u_y(s, 0) ds \quad (2.3.15)$$

$$\int_{\Gamma_3} v u_y ds = \int_0^L v(L - s, L) u_y(L - s, L) ds \quad (2.3.16)$$

apply change of variable $r = L - s$ on (2.3.16), and since v and u_y satisfy PBCs,

$$\begin{aligned} \int_{\Gamma_3} v u_y ds &= \int_0^L v(L - s, L) u_y(L - s, L) ds = - \int_L^0 v(r, L) u_y(r, L) dr \\ &= \int_0^L v(r, L) u_y(r, L) dr = \int_0^L v(s, L) u_y(s, L) ds = \int_{\Gamma_1} v u_y ds \end{aligned}$$

so,

$$\int_{\Gamma_3} v u_y ds - \int_{\Gamma_1} v u_y ds = 0 \quad (2.3.17)$$

In the same manner,

$$\int_{\Gamma_2} v u_x ds = \int_0^L v(L, s) u_x(L, s) ds \quad (2.3.18)$$

$$\int_{\Gamma_4} v u_x ds = \int_0^L v(0, L - s) u_x(0, L - s) ds \quad (2.3.19)$$

apply the change of variable $r = L - s$ on (2.3.19), and since u_x and v satisfy PBCs,

$$\begin{aligned} \int_{\Gamma_4} v u_x ds &= \int_0^L v(0, L - s) u_x(0, L - s) ds = - \int_L^0 v(0, r) u_x(0, r) dr \\ &= \int_0^L v(0, r) u_x(0, r) dr = \int_0^L v(0, s) u_x(0, s) ds = \int_{\Gamma_2} v u_x ds \end{aligned}$$

so,

$$- \int_{\Gamma_4} v u_x ds + \int_{\Gamma_2} v u_x ds = 0 \quad (2.3.20)$$

By (2.3.17) and (2.3.20),

$$\int_{\Gamma} \vec{v} \cdot \vec{V}(u) v ds = - \int_{\Gamma_1} v u_y ds + \int_{\Gamma_2} v u_x ds + \int_{\Gamma_3} v u_y ds - \int_{\Gamma_4} v u_x ds = 0$$

so, (2.3.14) implies,

$$\int_{\Omega} \nabla u \nabla v dx dy = \int_{\Omega} -\operatorname{div}(\nabla u) v dx dy$$

Therefore, (2.3.12) implies:

$$\int_{\Omega} \nabla u \nabla v \, dx \, dy + \int_{\Omega} uv \, dx \, dy = \int_{\Omega} wv \, dx \, dy$$

Thus the following variational formulation is obtained:

$$\underbrace{\langle \nabla u, \nabla v \rangle_{L^2(\Omega)} + \langle u, v \rangle_{L^2(\Omega)}}_{A(u,v)} = \underbrace{\langle w, v \rangle_{L^2(\Omega)}}_{F(v)} \quad (2.3.21)$$

Equivalent to:

$$\underbrace{\langle u, v \rangle_{H^1(\Omega)}}_{A(u,v)} = \underbrace{\langle w, v \rangle_{L^2(\Omega)}}_{F(v)} \quad (2.3.22)$$

2.3.3 Semi-Variational Formulation

1. Strong Semi-Variational Formulation

System (2.2.4) can be put now in the following strong semi-variational form (on the space variables) whereby one seeks for $u \in C([0, T]; H^2(\Omega) \cap H_p^1(\Omega)) \cap C^1((0, T); L^2(\Omega))$ and $w \in C([0, T], H_p^1(\Omega)) \cap C^1((0, T), L^2(\Omega))$ such that $\forall v \in H_p^1(\Omega), \forall t \in (0, T]$:

$$\begin{cases} \langle w_t, v \rangle_{L^2(\Omega)} + \langle \vec{V}(u) \cdot \nabla w, v \rangle_{L^2(\Omega)} = \langle ku_y, v \rangle_{L^2(\Omega)}, \\ \langle u, v \rangle_{H^1(\Omega)} = \langle w, v \rangle_{L^2(\Omega)}, \\ u(0) = u_0 \in H_p^2(\Omega) \\ w(0) = u_0 - \Delta u_0 \in H_p^1(\Omega) \end{cases} \quad (2.3.23)$$

2. Weak Semi-Variational Formulation

System (2.3.23) can be put now in the following weak semi-variational form (on the space variables) [9] whereby one seeks for $u \in C(0, T, H^2(\Omega) \cap H_p^1(\Omega))$ and $w \in C([0, T], L^2(\Omega)) \cap C^1((0, T), L^2(\Omega))$ such that $\forall v \in H_p^1(\Omega), \forall t \in (0, T]$:

$$\begin{cases} \langle w_t, v \rangle_{L^2(\Omega)} - \langle \vec{V}(u) \cdot \nabla v, w \rangle_{L^2(\Omega)} = \langle ku_y, v \rangle_{L^2(\Omega)}, \\ \langle u, v \rangle_{H^1(\Omega)} = \langle w, v \rangle_{L^2(\Omega)}, \\ u(0) = u_0 \in H_p^2(\Omega) \\ w(0) = u_0 - \Delta u_0 \in L^2(\Omega) \end{cases} \quad (2.3.24)$$

2.3.4 Time Integral Formulation

Newton's method requires a formulation which is time-derivative w_t free. That's why we derive Time Integral Formulation.

By integrating the first equation in (2.3.24) over the temporal interval $[t, t + \tau]$, with $0 \leq t \leq T - \tau$, one reaches the following L^2 integral formulation .

We are seeking $(u, w) \in L^2(0, T; H^2(\Omega) \cap H_p^1(\Omega)) \times L^2(0, T; L^2(\Omega))$ such that $\forall v \in W_p^{1,\infty} \cap H_p^1(\Omega)$, $\forall t, \tau$ s.t $0 \leq t < t + \tau \leq T$, and $\forall s \in [t, t + \tau]$:

$$\begin{cases} \langle w(t + \tau) - w(t), v \rangle_{L^2(\Omega)} = \int_t^{t+\tau} \langle \vec{V}(u(s)) \nabla v, w \rangle_{L^2(\Omega)} + \langle ku_y(s), v \rangle_{L^2(\Omega)} ds \\ \langle u(s), v \rangle_{H^1(\Omega)} = \langle w(s), v \rangle_{L^2(\Omega)}, \\ u(0) = u_0 \in H^2(\Omega) \cap H_p^2(\Omega) \\ w(0) = u_0 - \Delta u_0 \in L^2(\Omega) \end{cases} \quad (2.3.25)$$

2.4 Discretization

In this section, we start first using a finite-element space discretization to obtain semi-discrete systems for the coupled variational Hasegawa-Mima model (2.3.24).

2.4.1 Finite Element Space Semi-Discretization

We begin by constructing \mathbb{P}_1 Finite-element spaces on the square $\Omega = (0, L) \times (0, L)$.

Let $\mathbb{X} = \{x_i | i = 1, \dots, n\}$ be a partition of $[0, L]$ in the x - direction:

$$0 = x_1 < x_2 < \dots < x_n = L$$

and $\mathbb{Y} = \{y_j | j = 1, \dots, n\}$ be a similar partition of $[0, L]$ in y - direction:

$$0 = y_1 < y_2 < \dots < y_n = L.$$

Consequently, let \mathcal{N} be set of nodes associated with the conformal triangulation $\bar{\Omega}$:

$$\mathcal{N} = \{P_k | 1 \leq k \leq N = n^2\} = \{(x_i, y_j) | 1 \leq i, j \leq n\} = \mathbb{X} \times \mathbb{Y}.$$

Thus Ω is “meshed” using the resulting triangulation, based on set of nodes \mathcal{N} .

Practically, we generate such conformal finite-element using **Free-Fem++** [6] on a rectangle, obtaining:

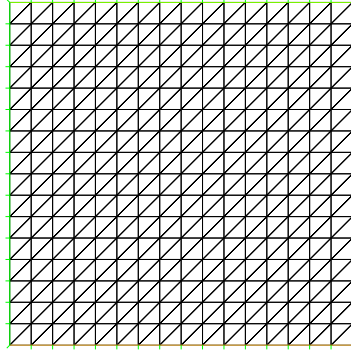


Figure 2.2: A conformal finite element meshing of the rectangle ($n \times n$)

This meshing generates a set of triangles E , $E = \{T_i | 1 \leq i \leq M\}$, such that:

- $T_i \cap T_j = \begin{cases} \text{triangle itself when } i = j \\ \text{vertex} \\ \text{one side} \\ \emptyset \end{cases}$
- $\bar{\Omega} = \bigcup_{i=1}^M \bar{T}_i$

This allows us defining \mathbb{P}_1 finite-element spaces.

Definition of $S_N^1(\Omega)$

\mathbb{P}_1 is the set of all polynomials of degree 1, in x and y . The space of \mathbb{P}_1 -finite elements on Ω is defined as follows:

$$S_N^1(\Omega) = \{\Phi \in C(\bar{\Omega}) | \Phi|_{T_i} \in \mathbb{P}_1, \forall i = 1, \dots, M\},$$

with the approximation property:

$$H^1(\Omega) = \overline{\bigcup_{N \geq 1} S_N^1(\Omega)},$$

i.e., $\forall v \in H^1(\Omega)$, its interpolant $\pi_N(v) \in S_N^1(\Omega)$ is such that:

$$\lim_{N \rightarrow \infty} \|v - \pi_N(v)\| = 0.$$

To obtain a Lagrangian basis for $S_N^1(\Omega)$, let $\Phi \in S_N^1(\Omega)$. For $(x, y) \in T \in E$, $\Phi(x, y) \in \mathbb{P}_1$. Then on every triangle $T = P_1P_2P_3$:

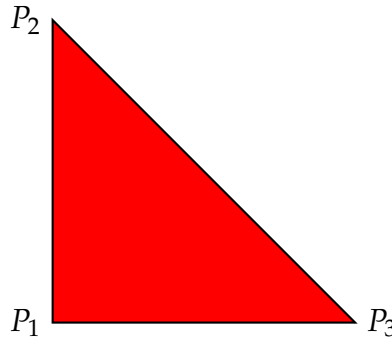


Figure 2.3: Right Triangle in Ω

:

$$\Phi(x, y) = b_T + a_T^1 x + a_T^2 y$$

Given the continuity of Φ on $\overline{\Omega}$, the dependence of Φ on 3 parameters allows writing:

$$\Phi(x, y) = \phi(P_1)\Psi_{P_1,T}(x, y) + \phi(P_2)\Psi_{P_2,T}(x, y) + \phi(P_3)\Psi_{P_3,T}(x, y),$$

where: $\Psi_{P_i,T}(x, y) = \alpha_i x + \beta_i y + \gamma_i$ for $i = \{1, 2, 3\}$.

The basis $\{\psi_{P_1,T}, \psi_{P_2,T}, \psi_{P_3,T}\}$ verifies

$$\Psi_{P_i,T}(P_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \text{ for } i, j = \{1, 2, 3\}.$$

So, for triangle $T \in E$, with vertices $\{P_i\}$, $\psi_{P_i,T}$ can be written as:

$$\Psi_{P_i,T}(x, y) = 1 - c_i(x - x_{P_i}) - d_i(y - y_{P_i}) \quad \text{for } i = \{1, 2, 3\}.$$

Let $E_p = \{T \in E | P \text{ is a vertex of } T\}$ be the set of triangles from E having $P \in \mathcal{N}$ as a vertex. Then for $(x, y) \in \Omega$, $\psi_P(x, y) = 0$, if $\forall T \in E_p, (x, y) \notin T$ and $\psi_P(x, y) = \psi_{P,T}(x, y)$ if $(x, y) \in T$ where $T \in E_p$. Therefore:

$$\begin{aligned} S_N^1(\Omega) &= \text{span}\{\psi_P(x, y) | P \in \mathcal{N} \text{ and } N \text{ is the total number of nodes}\} \\ &= \text{span}\{\psi_{P_i}(x, y) | 1 \leq i \leq N\}, \end{aligned}$$

where $\psi_P(x, y)$ is the basis function associated with node P . Note that the support of ψ_P :

$$\text{supp}(\psi_P) = \bigcup_{T \in E_P} T,$$

i.e.

$$\begin{cases} \psi_P(x, y) = 0 & (x, y) \notin \bigcup_{T \in E_P} T \\ \psi_P(x, y) = 1 & (x, y) \in \bigcup_{T \in E_P} T \end{cases}$$

Hence $\Phi(x, y) = \sum_{P \in \mathcal{N}} \Phi(P) \psi_P(x, y) = \sum_{i=1}^N \Phi(P_i) \psi_{P_i}(x, y)$.

Therefore to any $\Phi \in S_N^1(\Omega)$, one associate uniquely:

$$V = \begin{pmatrix} \Phi(P_1) \\ \vdots \\ \Phi(P_{n^2}) \end{pmatrix} \in \mathbb{R}^{n^2},$$

where $N = n^2$ is the total number of nodes. This implies that finding a function $\Phi \in S_N^1(E)$ reduces to finding its values at the nodes \mathcal{N} . Thus given a triangulation $E = \{T_i | 1 \leq i \leq M\}$ that covers Ω in a conformal way, implementing the finite element-method requires two basic data structures: Nodes and Elements. As mentioned $\bigcup_{N \geq 1} S_N^1(\Omega)$ is an approximation basis in $H^1(\Omega)$. Let $B_N = \{\phi_{P_i} | 1 \leq i \leq N\}$ be such finite-element basis of functions with compact support in Ω .

For $v \in H^1(\Omega)$, we can define $\pi_N(v) := v_N \in S_N^1(\Omega)$ to be the interpolant of v in $S_N^1(\Omega)$:

$$v_N(x, y) = \sum_{i=1}^N V_{P_i} \phi_{P_i}(x, y), \quad V_{P_i} = v_N(x_{P_i}, y_{P_i})$$

These auxiliary structures allow to generate discrete systems used to obtain the pair $\{u_N, w_N\}$ of finite element approximations to the coupled Elliptic-Hyperbolic system (2.3.24).

Definition of $S_{N,P}^1(\Omega)$

Define $S_{N,P}^1(\Omega)$ as follows:

$$S_{N,P}^1(\Omega) = \{v \in S_N^1(\Omega) | v \text{ satisfies PBC on } \Gamma\}.$$

Knowing that $S_N^1(\Omega) = \{\psi_{P_i}, 1 \leq i \leq N\}$, to find a basis for $S_{N,P}^1(\Omega)$, we proceed as follows. Given the square $\Omega = (0, L) \times (0, L)$:

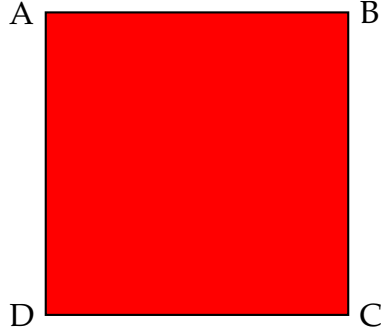


Figure 2.4: Domain Ω

we let N_U be the set of interior nodes of Ω , and N_K be the set of nodes on $\partial\Omega = \Gamma$. Divide N_K the set of nodes on boundary Γ to 2 parts N_{KU_1} and N_{KU_2} where:

$$N_{KU_1} = \{I \in \Gamma : | I \in [AB] \text{ or } [BC]\} = \{I \in \Gamma : |x(I) = L \text{ or } y(I) = L\}$$

$$N_{KU_2} = \{J \in \Gamma : | J \in [AD] \text{ or } [DC]\} = \{I \in \Gamma : |x(I) = 0 \text{ or } y(I) = 0\}$$

Note that $u_N \in S_{N,p}^1(\Omega)$, then u_N satisfies PBC, i.e.,

$$u_N(0, y) = u_N(L, y), \forall y \in [0, L]; u_N(x, 0) = u_N(x, L), \forall x \in [0, L].$$

Therefore, if we let $I \in N_{KU_1}$ then there exists $J \in N_{KU_2}$ such that: $u_N(I) = u_N(J)$. Conversely, let $I \in N_{KU_2}$ then there exists $J \in N_{KU_1}$ such that $u_N(I) = u_N(J)$.

Therefore our set of unknowns will be reduced to $N_1 = n^2 - (2n - 1) = (n - 1)^2$ equations instead of $N = n^2$ equations, where N is the total number of nodes. Thus:

$$\begin{aligned} u_N &= \sum_{I \in \mathcal{N}} U_I \psi_I \\ &= \sum_{I \in N_U} U_I \psi_I + \sum_{I \in N_{KU_1}} U_I \psi_I + \sum_{J \in N_{KU_2}} U_J \psi_J \\ &= \sum_{I \in N_U} U_I \psi_I + \sum_{I \in N_{KU_1}} U_I (\psi_I + \psi_J) \text{ (where } J \in N_{KU_2} \text{ such that } u_N(I) = u_N(J)) \\ &= \sum_{I \in N_U} U_I \psi_I + \sum_{I \in N_{KU_1}} U_I \tilde{\psi}_I \end{aligned}$$

Define now:

$$\{\phi_I | I \in N_U \cup N_{KU_1}\} = \{\psi_I | I \in N_U\} \cup \{\tilde{\psi}_I | I \in N_{KU_1}\} = \{\phi_{I_i} | 1 \leq I_i \leq N_1\}.$$

Therefore, $S_{N,P}^1(\Omega) = \text{span}\{\phi_{I_i}, \text{ where } 1 \leq i \leq N_1\}$ and:

$$v_N \in S_{N,P}^1(\Omega) \iff v_N(x, y) = \sum_{i=1}^{N_1} V_{I_i} \phi_{I_i}(x, y),$$

where $v_N(I_i) = V_{I_i}, 1 \leq i \leq N_1$.

For $v \in H_P^1(\Omega)$, $\pi_N(v) := \sum_{i=1}^{N_1} V_{I_i} \phi_{I_i} \in S_{N,P}^1(\Omega)$ to be the interpolant of v in $S_{N,P}^1(\Omega)$ where $V_{I_i} = v(I_i)$. Then the approximation property of $S_{N,P}^1(\Omega)$ in $H_P^1(\Omega)$ can be stated as follows:

$$\forall v \in H_P^1(\Omega) \quad \lim_{N \rightarrow \infty} \|v - \pi_N(v)\| = 0.$$

2.4.2 Full Discretization

For the time Discretization, we define a set nodes of $[0, T]$:

$$\mathcal{T} = \{t_i | i = 0, \dots, m\}$$

such that :

- (a) $t_i \neq t_j$ for $i \neq j$
- (b) $t_0 = 0$ and $t_m = T$
- (c) $t_i - t_{i-1} = \tau$, where $i = 1, \dots, m$.

To discretize (2.3.24), we start with $u_N(0) = \pi_N(u_0)$ and $w_N(0) = \pi_N(w_0)$, then given

$$u_N(t) = \pi_n(u) = \sum_{i=1}^{N_1} U_{I_i}(t) \phi_{I_i} \text{ and } w_N(t) = \pi_n(w) = \sum_{i=1}^{N_1} W_{I_i}(t) \phi_{I_i} \in S_P^1(\Omega) \in S_{N,P}^1(\Omega),$$

where $u_N(x_{I_i}, y_{I_i}, t) = U_{I_i}(t)$, $1 \leq i \leq N_1$ and $w_N(x_{I_i}, y_{I_i}, t) = W_{I_i}(t)$, $1 \leq i \leq N_1$, we seek $u_N(t + \tau) \in S_{N,P}^1(\Omega)$ and $w_N(t + \tau) \in S_{N,P}^1(\Omega)$ such that:

$$\left\{ \begin{array}{l} \langle w_N(t + \tau) - w_N(t), v \rangle_{L^2(\Omega)} = \dots \\ \dots \int_t^{t+\tau} \langle \vec{V}(u) \nabla v, w \rangle_{L^2(\Omega)} + \langle k u_{N,y}, v \rangle_{L^2(\Omega)} ds \\ \langle u_N(s), v \rangle_{H^1(\Omega)} = \langle w_N(s), v \rangle_{L^2(\Omega)}, \\ \forall v \in S_{N,P}^1(\Omega) \\ \forall t, 0 \leq t < t + \tau \leq T, \quad \text{and } \forall s \in [t, t + \tau]. \end{array} \right. \quad (2.4.1)$$

To fully discretize of Hasegawa Mima equation,

$$\int_t^{t+\tau} \langle \vec{V}(u(s)) \nabla v, w(s) \rangle_{L^2(\Omega)} + \langle k u_{N,y}(s), v \rangle_{L^2(\Omega)} ds$$

is first discretized using an implicit right rectangular rule $\int_a^b f(s) ds \approx (b - a)f(b)$.

If we let $f(s) = \langle \vec{V}(u(s))\nabla v, w(s) \rangle_{L^2(\Omega)} + \langle ku_{N,y}(s), v \rangle_{L^2(\Omega)}$, $a = \tau$, $b = t + \tau$, then:

$$\begin{aligned} I &= \int_t^{t+\tau} \langle \vec{V}(u(s))\nabla v, w(s) \rangle_{L^2(\Omega)} + \langle ku_{N,y}(s), v \rangle_{L^2(\Omega)} ds \\ &\approx \tau(\langle \vec{V}(u(t + \tau))\nabla v, w(t + \tau) \rangle_{L^2(\Omega)} + \langle ku_{N,y}(t + \tau), v \rangle_{L^2(\Omega)}) \end{aligned}$$

Therefore, starting from (u_0, w_0) , then given $(u_N(t), w_N(t)) \in S_{N,P}^1 \times S_{N,P}^1(\Omega)$, we seek $(u_N(t + \tau), w_N(t + \tau)) \in S_{N,P}^1 \times S_{N,P}^1(\Omega)$ such that:

$$\left\{ \begin{array}{l} \langle w_N(t + \tau) - w_N(t), v \rangle_{L^2(\Omega)} = \dots \\ \dots \tau \langle \vec{V}(u(t + \tau))\nabla v, w \rangle_{L^2(\Omega)} + \tau \langle ku_{N,y}(t + \tau), v \rangle_{L^2(\Omega)} \\ \langle u_N(s), v \rangle_{H^1(\Omega)} = \langle w_N(s), v \rangle_{L^2(\Omega)}, \\ \forall v \in S_{N,P}^1(\Omega) \\ \forall s \in [t, t + \tau] \end{array} \right. \quad (2.4.2)$$

Using the expressions:

$u_N(t) = \sum_{i=1}^N U_{I_i}(t)\phi_{I_i} \in S_{N,P}^1(\Omega)$, $U_{I_i}(t) = u_N(x_{I_i}, y_{I_i}, t)$ and $w_N(t) = \sum_{i=1}^N W_{I_i}(t)\phi_{I_i} \in S_{N,P}^1(\Omega)$, $W_{I_i}(t) = w_N(x_{I_i}, y_{I_i}, t)$, define the vectors: $W(t) = \{W_{I_i}(t)\}_{1 \leq i \leq N_1} \in \mathbb{R}^{N_1}$ and $U(t) = \{U_{I_i}(t)\}_{1 \leq i \leq N_1} \in \mathbb{R}^{N_1}$ so that (3.1.1) can be written as follows: Given vectors $\{U(t), W(t)\} \in \mathbb{R}^{N_1} \times \mathbb{R}^{N_1}$, one seeks $\{U(t + \tau), W(t + \tau)\} \in \mathbb{R}^{N_1} \times \mathbb{R}^{N_1}$ such that

$$\left\{ \begin{array}{l} (M + \tau S(U + \tau))W(t + \tau) - \tau R U(t + \tau) = M W(t) \\ K U(s) = M W(s) \quad \forall s \in \{t, t + \tau\} \end{array} \right. \quad (2.4.3)$$

where $M, S(U), R, K$ are $N_1 \times N_1$ matrices defined as follows:

$$M = \{\langle \phi_{I_i}, \phi_{I_j} \rangle_{L^2(\Omega)} \mid 1 \leq i, j \leq N_1\}$$

$$K = \{\langle \phi_{I_i}, \phi_{I_j} \rangle_{H^1(\Omega)} \mid 1 \leq i, j \leq N_1\}$$

$$R = \{\langle k\phi_{I_i,y}, \phi_{I_j} \rangle_{L^2(\Omega)} \mid 1 \leq i, j \leq N_1\}$$

$$S(U) = \{-\langle \vec{V}(u_N) \cdot \nabla \phi_{I_i}, \phi_{I_j} \rangle_{L^2(\Omega)} \mid 1 \leq i, j \leq N_1\}$$

with: $u_N(t) = \sum_{k=1}^N U_{I_k}(t)\phi_{I_k}$

and

$$\begin{aligned}
- \langle \vec{V}(u_N) \cdot \nabla \phi_{I_j}, \phi_{I_i} \rangle_{L^2(\Omega)} &= - \int_{\Omega} -\phi_{I_i} \frac{\partial \phi_{I_j}}{\partial x} \sum_{k=1}^N U_{I_k}(t) \frac{\partial \phi_{I_k}}{\partial y} + \phi_{I_i} \frac{\partial \phi_{I_j}}{\partial y} \sum_{k=1}^N U_{I_k}(t) \frac{\partial \phi_{I_k}}{\partial x} dx dy \\
&= \int_{\Omega} \phi_{I_i} \frac{\partial \phi_{I_j}}{\partial x} \sum_{k=1}^N U_{I_k}(t) \frac{\partial \phi_{I_k}}{\partial y} - \phi_{I_i} \frac{\partial \phi_{I_j}}{\partial y} \sum_{k=1}^N U_{I_k}(t) \frac{\partial \phi_{I_k}}{\partial x} dx dy \\
&= \langle \vec{V}(u_N) \cdot \nabla \phi_{I_i}, \phi_{I_j} \rangle_{L^2(\Omega)}
\end{aligned}$$

Here M , K and R constant matrices over time, and $S(U)$ changing with time. The coupled system obtained in (3.1.2) (highly implicit and non linear) is used to advance the pair $(U(s), W(s))$, starting from $(U(0), W(0))$ and computing successively:

$$(U(\tau), W(\tau)), (U(2\tau), W(2\tau)) \dots (U(m\tau), W(m\tau)).$$

Chapter 3

Solving Fully Discrete Hasegawa-Mima System

In this section, we discuss Newton type methods (Full Newton, Quasi Newton, Modified Newton) to solve the full discrete system of Hasegawa-Mima model and present for each variant of Newton's Method the corresponding algorithm which is implemented using Free-FEM++. For simplicity, in the rest of the thesis we replace N_1 by N .

3.1 Formulation of Newton's method

From (3.1.2), one has the following non-linear problem. Given vectors $(U(t), W(t)) \in \mathbb{R}^N \times \mathbb{R}^N$, we seek $(U(t + \tau), W(t + \tau)) \in \mathbb{R}^N \times \mathbb{R}^N$ such that:

$$\begin{cases} (M + \tau S(U + \tau))W(t + \tau) - \tau RU(t + \tau) = MW(t) \\ KU(s) = MW(s) \quad \forall s \in \{t, t + \tau\} \end{cases} \quad (3.1.1)$$

This is equivalent to find $(U, W) \in \mathbb{R}^N \times \mathbb{R}^N$ such that:

$$\begin{cases} (M + \tau S(U))W - \tau RU - Z = 0 \\ KU - MW = 0 \end{cases} \quad (3.1.2)$$

where $U = U(t + \tau)$, $W = W(t + \tau)$, and $Z = MW(t)$ (given). Define F_1 and F_2 :

$$\begin{aligned} F_1: \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \\ (U, W) &\rightarrow F_1(U, W) = (M + \tau S(U))W - \tau RU - Z \\ F_2: \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \\ (U, W) &\rightarrow F_2(U, W) = KU - MW. \end{aligned}$$

Solving (3.1.2) equivalent to find $(U, W) \in \mathbb{R}^N \times \mathbb{R}^N$ such that:

$$\begin{cases} F_1(U, W) = 0 \\ F_2(U, W) = 0 \end{cases} \quad (3.1.3)$$

Define

$$\begin{aligned} F: \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \times \mathbb{R}^N \\ (U, W) &\rightarrow [F_1(U, W) \quad F_2(U, W)]^T. \end{aligned} \quad (3.1.4)$$

Thus we look for solving $F(U, W) = 0$ using Newton's Method.

Assuming that $(U(t), W(t))$ was already computed/approximated, then $(U(t + \tau), W(t + \tau))$ is approximated using Newton's method by solving the following iteratively with $(U^{(0)}, W^{(0)}) = (U(t), W(t))$

$$J_F(U^{(0)}, W^{(0)}) \begin{bmatrix} U^{(1)} - U^{(0)} \\ W^{(1)} - W^{(0)} \end{bmatrix} = -F(U^{(0)}, W^{(0)}) \quad (3.1.5)$$

The process is repeated as follows until convergence is reached up to a sufficient precise value.

$$J_F(U^{(k)}, W^{(k)}) \begin{bmatrix} U^{(k+1)} - U^{(k)} \\ W^{(k+1)} - W^{(k)} \end{bmatrix} = -F(U^{(k)}, W^{(k)}) \quad (3.1.6)$$

$$\iff J_F(U^{(k)}, W^{(k)}) \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} = J_F(U^{(k)}, W^{(k)}) \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} - F(U^{(k)}, W^{(k)}) \quad (3.1.7)$$

where $J_F(U, W)$ is the Jacobian of $F(U, W)$

Let's compute the Jacobian:

$$J_F(U, W) = \begin{bmatrix} F_{1,U} & F_{1,W} \\ F_{2,U} & F_{2,W} \end{bmatrix} = \begin{bmatrix} \tau(S(U)W)_U - \tau R & M + \tau S(U) \\ K & -M \end{bmatrix}$$

It remains to find the derivative of $S(U)W$ with respect to U , denoted by $(S(U)W)_U$.

Given that the matrix $S(U)$ is linear in U where

$$U = \begin{bmatrix} U_{I_1} \\ U_{I_2} \\ \vdots \\ \vdots \\ U_{I_N} \end{bmatrix} = \sum_{j=1}^N U_{I_j} e_j,$$

then, $S(U) = S(\sum_{j=1}^N U_{I_j} e_j) = \sum_{j=1}^N U_{I_j} S(e_j)$

then, $S(U)W = \sum_{j=1}^{N_1} U_{I_j} S(e_j)W = U_{I_1} S(e_1)W + U_{I_2} S(e_2)W + \dots + U_{I_N} S(e_N)W$

so,

$$B(W) = (S(U)W)_U = \begin{bmatrix} S(e_1)W & S(e_2)W & \dots & S(e_N)W \end{bmatrix}.$$

$$\text{Hence, } J_F(U, W) = \begin{bmatrix} \tau B(W) - \tau R & M + \tau S(U) \\ K & -M \end{bmatrix}.$$

$$\text{Replace } J_F(U^{(k)}, W^{(k)}) = \begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \text{ in (3.1.7),}$$

hence one will have to solve the following :

$$\begin{aligned} \begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} \\ &\quad - F(U^{(k)}, W^{(k)}) \\ \iff \begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} \\ &\quad - \begin{bmatrix} (M + \tau S(U^{(k)}))W^{(k)} - \tau R U^{(k)} - Z \\ K U^{(k)} - M W^{(k)} \end{bmatrix} \end{aligned}$$

Using the following identity:

$$B(W)U = \begin{bmatrix} S(e_1)W & S(e_2)W & \dots & S(e_N)W \end{bmatrix} \begin{bmatrix} U_{I_1} \\ U_{I_2} \\ \vdots \\ U_{I_N} \end{bmatrix} = \sum_{j=1}^N U_{I_j} S(e_j)W = S(U)W \quad (3.1.8)$$

Thus,

$$\begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tau S(U^{(k)})W^{(k)} + Z \\ 0 \end{bmatrix} \quad (3.1.9)$$

Computation $B = B(W^k)$

Note that $B(W^{(k)}) = \begin{bmatrix} S(e_1)W^{(k)} & S(e_2)W^{(k)} & \dots & S(e_N)W^{(k)} \end{bmatrix}$ should be computed at every Newton iteration, where e_i 's are the canonical basis of \mathbb{R}^N .

Jacobian matrix $J = J(U^k, W^k)$

Jacobian matrix is $2N \times 2N$ block matrix

$$\begin{pmatrix} J_1 & J_2 \\ J_3 & J_4 \end{pmatrix}$$

where

$$J_1 = \tau B(W^k) - \tau R$$

$$J_2 = M + \tau S(U^k)$$

$$J_3 = K$$

$$J_4 = -M$$

Note that J_1 and J_2 should be computed at every Newton iteration.

Algorithm 1 Solving HM using Newton's Method

Input: A : stiffness matrix ; M : Mass matrix; $K = M + A$; $S(U)$; R : As defined in [9]; $B(W)$: Matrix defined above; J : block matrix defined above ; U_0 ; W_0 : the discrete initial condition vectors; T : end time; τ : time step; k_{max} : maximum Newton iterations; N : total number of nodes

Output: U : matrix containing the computed vectors U_t for $t = 0, 1, \dots, T$

```

1: for  $t = 0 : \tau : T$  do
2:    $U_{t,0} = U_t$ ;  $W_{t,0} = W_t$ ;  $error = 1$ ;  $k = 0$ 
3:   Let  $Z = M * W_t$ ;
4:   while ( $error > tol$  and  $k < k_{max}$ ) do
5:     Let  $g = \tau S(U_{t,k}) * W_{t,k} + Z$ ;
6:     Let  $r(0 : N - 1) = g$  and  $r(n : 2 * N - 1) = 0$ ;
7:     Let  $J_1 = \tau B(W_{t,k}) - \tau R$ ,  $J_2 = \tau S(U_{t,k}) + M$ ;
8:     Construct  $J$ :  $J = [[J_1, J_2], [K, -M]]$ ;
9:     Solve for  $V$ :  $J * V = r$ ;
10:    let  $U_{t,k+1} = V(0 : N - 1)$  and  $W_{t,k+1} = V(n : 2 * N - 1)$ ;
11:     $error = \frac{\|U_{t,k+1} - U_{t,k}\|}{\|U_{t,k}\|}$ ;
12:     $k = k + 1$ ;
13:  end while
14:   $U_{t+1} = U_{t,k}$ ;
15:   $W_{t+1} = W_{t,k}$ ;
16: end for

```

3.2 Variants of Newton's Method

1. Chord's Method

Given vectors $\{U(t), W(t)\} \in \mathbb{R}^N \times \mathbb{R}^N$
 one seeks $\{U(t + \tau), W(t + \tau)\} \in \mathbb{R}^N \times \mathbb{R}^N$ such that

$$\begin{cases} (M + \tau S(U + \tau))W(t + \tau) - \tau RU(t + \tau) = MW(t) \\ KU(s) = MW(s) \quad \forall s \in \{t, t + \tau\} \end{cases} \quad (3.2.1)$$

This equivalent to find $(U, W) \in \mathbb{R}^{N^1} \times \mathbb{R}^N$ such that

$$\begin{cases} (M + \tau S(U))W - \tau RU - Z = 0 \\ KU - MW = 0 \end{cases} \quad (3.2.2)$$

where $U = U(t + \tau)$, $W = W(t + \tau)$, and $Z = MW(t)$ (given)

Define F_1 and F_2 :

$$\begin{aligned} F_1: \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \\ (U, W) &\mapsto F_1(U, W) = (M + \tau S(U))W - \tau RU - Z \\ F_2: \mathbb{R}^N \times \mathbb{R}^{N^1} &\rightarrow \mathbb{R}^N \\ (U, W) &\mapsto F^2(U, W) = KU - MW \end{aligned}$$

Solving (3.1.2) is equivalent to find $\{U, V\} \in \mathbb{R}^N \times \mathbb{R}^N$ such that:

$$\begin{cases} F_1(U, W) = 0 \\ F_2(U, W) = 0 \end{cases} \quad (3.2.3)$$

Define

$$\begin{aligned} F: \mathbb{R}^N \times \mathbb{R}^N &\rightarrow \mathbb{R}^N \times \mathbb{R}^N \\ (U, W) &\mapsto [F_1(U, W) \quad F_2(U, W)]^T \end{aligned}$$

We will solve $F(U, W) = 0$ using chord's method.

$$\begin{aligned} J_F(U^{(0)}, W^{(0)}) \begin{bmatrix} U^{(k+1)} - U^{(k)} \\ W^{(k+1)} - W^{(k)} \end{bmatrix} &= -F(U^{(k)}, W^{(k)}) \\ \iff J_F(U^{(0)}, W^{(0)}) \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= J_F(U^{(0)}, W^{(0)}) \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} - F(U^{(k)}, W^{(k)}) \end{aligned} \quad (3.2.4)$$

where $U^{(0)} = U(t)$, $W^{(0)} = W(t)$, and $J_F(U, W)$ is the Jacobian of $F(U, W)$:

$$J_F(U, W) = \begin{bmatrix} \tau B(W) - \tau R & M + \tau S(U) \\ K & -M \end{bmatrix} \text{ (proved)}$$

Replace $J_F(U^{(0)}, W^{(0)}) = \begin{bmatrix} \tau B(W^{(0)}) - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix}$ in (3.2.4),

Denote $B(W^{(0)}) = B^{(0)}$, hence one has to solve the following:

$$\begin{aligned} \begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} \\ &\quad - F(U^{(k)}, W^{(k)}) \\ \Leftrightarrow \begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k)} \\ W^{(k)} \end{bmatrix} \\ &\quad - \begin{bmatrix} (M + \tau S(U^{(k)}))W^{(k)} - \tau R U^{(k)} - Z \\ KU^{(k)} - MW^{(k)} \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \tau B^{(0)} U^{(k)} - \tau R U^{(k)} + (M + \tau S(U^{(0)}))W^{(k)} \\ KU^{(k)} - MW^{(k)} \end{bmatrix} \\ &\quad - \begin{bmatrix} (M + \tau S(U^{(k)}))W^{(k)} - \tau R U^{(k)} - Z \\ KU^{(k)} - MW^{(k)} \end{bmatrix} \end{aligned}$$

Thus,

$$\begin{bmatrix} \tau B^{(0)} - \tau R & M + \tau S(U^{(0)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tau B^{(0)} U^{(k)} + \tau S(U^{(0)})W^{(k)} - \tau S(U^{(k)})W^{(k)} + Z \\ 0 \end{bmatrix} \quad (3.2.5)$$

Computation of matrix $B = B(W^0)$

$B(W^{(0)}) = [S(e_1)W^{(0)} \quad S(e_2)W^{(0)} \quad \dots \quad S(e_{N_1})W^{(0)}]$ is computed once per time iteration before Chord's iterations, where e_i 's are the canonical basis of R^N

Jacobian matrix $J = J(U^0, W^0)$

Jacobian matrix is $2N \times 2N$ block matrix is computed once per time it-

eration before Chord's iterations. $\begin{pmatrix} J_1 & J_2 \\ J_3 & J_4 \end{pmatrix}$

where

$$J_1 = \tau B(W^0) - \tau R$$

$$J_2 = M + \tau S(U^0)$$

$$J_3 = K$$

$$J_4 = -M$$

Algorithm 2 Solving HM using Chord's Method

Input: A : stiffness matrix ; M : Mass matrix; $K = M + A$; $S(U)$; R : As defined in [9]; $B(W)$: Matrix defined above; J : block matrix defined above ; $U_0; W_0$: the discrete initial condition vectors; T : end time; τ : time step; k_{max} : maximum Newton iterations; N : total number of nodes

Output: U : matrix containing the computed vectors U_t for $t = 0, 1, \dots, T$

```

1: for  $t = 0 : \tau : T$  do
2:    $U_{t,0} = U_t; W_{t,0} = W_t; error = 1; k = 0$ 
3:   Let  $Z = M * W_t$ ;
4:   Let  $J_1 = \tau B(W_{t,0}) - \tau R, J_2 = \tau S(U_{t,0}) + M$ ;
5:   Construct  $J$ :  $J = [[J_1, J_2], [K, -M]]$ ;
6:   while (  $error > tol$  and  $k < k_{max}$  ) do
7:     Let  $g = \tau B(W_{t,0})U_{t,k} + \tau S(U_{t,0}) * W_{t,k} - \tau S(U_{t,k}) * W_{t,k} + Z$ ;
8:     Let  $r(0 : N - 1) = g$  and  $r(n : 2 * N - 1) = 0$ ;
9:     Solve for  $V$ :  $J * V = r$ ;
10:    let  $U_{t,k+1} = V(0 : N - 1)$  and  $W_{t,k+1} = V(n : 2 * N - 1)$ ;
11:     $error = \frac{\|U_{t,k+1} - U_{t,k}\|}{\|U_{t,k}\|}$ ;
12:     $k = k + 1$ ;
13:  end while
14:   $U_{t+1} = U_{t,k}$ ;
15:   $W_{t+1} = W_{t,k}$ ;
16: end for

```

2. Modified Newton's Method

Using the newton's equation

$$\begin{bmatrix} \tau B(W^{(k)}) - \tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tau S(U^{(k)})W^{(k)} + Z \\ 0 \end{bmatrix} \quad (3.2.6)$$

$$\Leftrightarrow \begin{cases} \tau B(W^{(k)})U^{(k+1)} - \tau RU^{(k+1)} + MW^{(k+1)} + \tau S(U^{(k)})W^{(k+1)} = \tau S(U^{(k)})W^{(k)} + Z \\ KU^{(k+1)} - MW^{(k+1)} = 0 \end{cases} \quad (3.2.7)$$

we need to get rid of the computation of the $B(W^{(k)})$ matrix and speed up the procedure.

From identity (3.1.8):

$$B(W^{(k)})U^{(k+1)} = S(U^{(k+1)})W^{(k)} \approx S(U^{(k)})W^{(k)}$$

$$\begin{cases} \tau S(U^{(k)})W^{(k)} - \tau RU^{(k+1)} + MW^{(k+1)} + \tau S(U^{(k)})W^{(k+1)} = \tau S(U^{(k)})W^{(k)} + Z \\ KU^{(k+1)} - MW^{(k+1)} = 0 \end{cases} \quad (3.2.8)$$

$$\Leftrightarrow \begin{cases} -\tau RU^{(k+1)} + MW^{(k+1)} + \tau S(U^{(k)})W^{(k+1)} = Z \\ KU^{(k+1)} - MW^{(k+1)} = 0 \end{cases} \quad (3.2.9)$$

$$\Leftrightarrow \begin{bmatrix} -\tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ W^{(k+1)} \end{bmatrix} = \begin{bmatrix} Z \\ 0 \end{bmatrix} \quad (3.2.10)$$

Computation of jacobian matrix J

$$J = \begin{bmatrix} -\tau R & M + \tau S(U^{(k)}) \\ K & -M \end{bmatrix}$$

Jacobian matrix is $2N \times 2N$ block matrix

$$\begin{pmatrix} J_1 & J_2 \\ J_3 & J_4 \end{pmatrix}$$

where

$$J_1 = -\tau R$$

$$J_2 = M + \tau S(U^k)$$

$$J_3 = K$$

$$J_4 = -M$$

J_2 will be computed at every Modified Newton iteration.

Algorithm 3 Solving HM using Modified Newton Method

Input: A : stiffness matrix ; M : Mass matrix; $K = M + A$; $S(U)$; R : As defined in [9]; J : block matrix defined above ; $U_0; W_0$: the discrete initial condition vectors; T : end time; τ : time step;

k_{max} : maximum Newton iterations; n : total number of nodes

Output: U : matrix containing the computed vectors U_t for $t = 0, 1, \dots, T$

```
1: for  $t = 0 : \tau : T$  do
2:    $U_{t,0} = U_t$ ;  $W_{t,0} = W_t$ ;  $error = 1$ ;  $k = 0$ 
3:   Let  $Z = M * W_t$ ;
4:   Let  $r(0 : N - 1) = Z$  and  $r(n : 2 * N - 1) = 0$ ;
5:   while (  $error > tol$  and  $k < k_{max}$  ) do
6:     let  $J_2 = M + \tau S(U_{t,k})$ ;
7:     Construct  $J$ :  $J = [[-\tau R, J_2], [K, -M]]$ ;
8:     Solve for  $V$ :  $J * V = r$ ;
9:     let  $U_{t,k+1} = V(0 : N - 1)$  and  $W_{t,k+1} = V(n : 2 * N - 1)$ ;
10:     $error = \frac{\|U_{t,k+1} - U_{t,k}\|}{\|U_{t,k}\|}$ ;
11:     $k = k + 1$ ;
12:   end while
13:    $U_{t+1} = U_{t,k}$ ;
14:    $W_{t+1} = W_{t,k}$ ;
15: end for
```

Chapter 4

Testing

In this chapter, we present the results of our numerical simulations implemented using Free-Fem++ software [6]. In section 4.1, we analyze and compare the solutions of the three methods for several initial data. Then, in section 4.2 we assess the efficiency of our Newton-type approach as compared to the semi-linear approach presented in [9].

4.1 Newton-Type Methods

We test three algorithms [1] (Newton's method), [2] (Chord's method) and [3] (Modified Newton's method) for $\frac{n_0}{w_{ci}} = e^{Ax+B}$, i.e $p = Ax + B$ where $p_x = A$ and $p_y = 0$. We consider different initial conditions for same exponential density profile n_0 . We analyze the following cases:

4.1.1 Test 1

We set the domain $\Omega = [0, \pi] \times [0, \pi]$ with $A = 12$. The initial condition $u_0(x, y) = 10^{-5} \sin(3x)$, $\tau = 0.05$, and $T = 50$. We test the algorithm for the case where $n_0 = 10^{19}e^{-x}$, and $w_{ci} = 10^7$.

In Figures 4.1, 4.2, and 4.3, we plot the solutions of Algorithms 1, 2 and 3 respectively at different times, where we consider number of intervals in the x and y directions to be $n = 16$. In Figures 4.4, and 4.5, we plot the solutions of Algorithm 3 at different times whereby we consider the number of intervals in x and y directions to be $n = 32$ and 64 respectively.

The same behavior is observed for the three methods where the solution is stationary and unchanging till $T = 50$, as shown in figures 4.1, 4.2,

4.3 for $n = 16$. Moreover, this behavior is not affected by the increase in the number of intervals n as shown in figures 4.3, 4.4 and 4.5 for the Modified Newton's method.

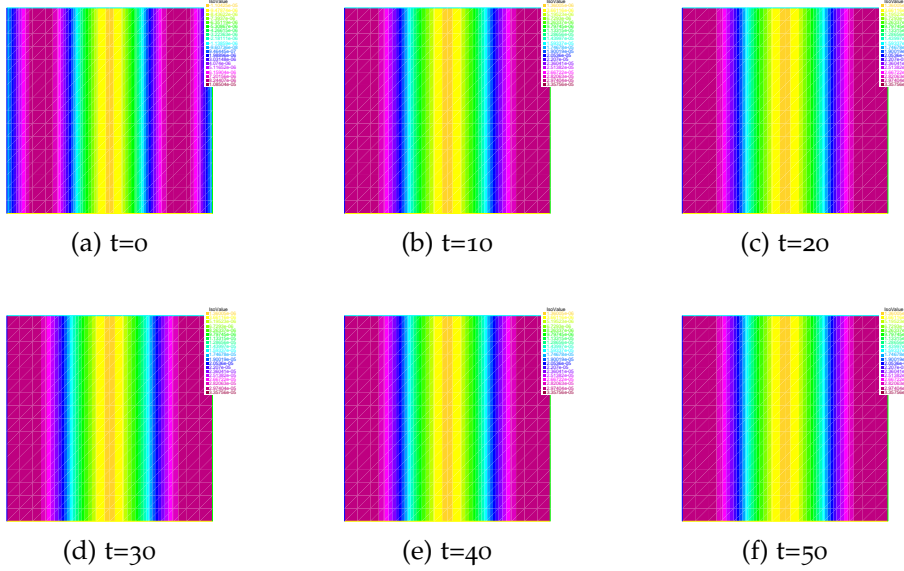


Figure 4.1: Time evolution of solution u using full newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

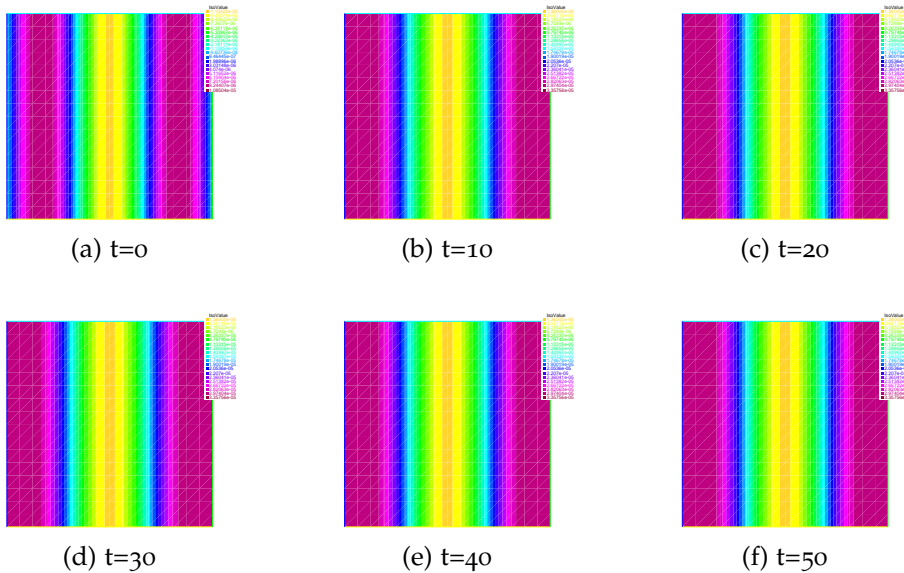


Figure 4.2: Time evolution of solution u using chord method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

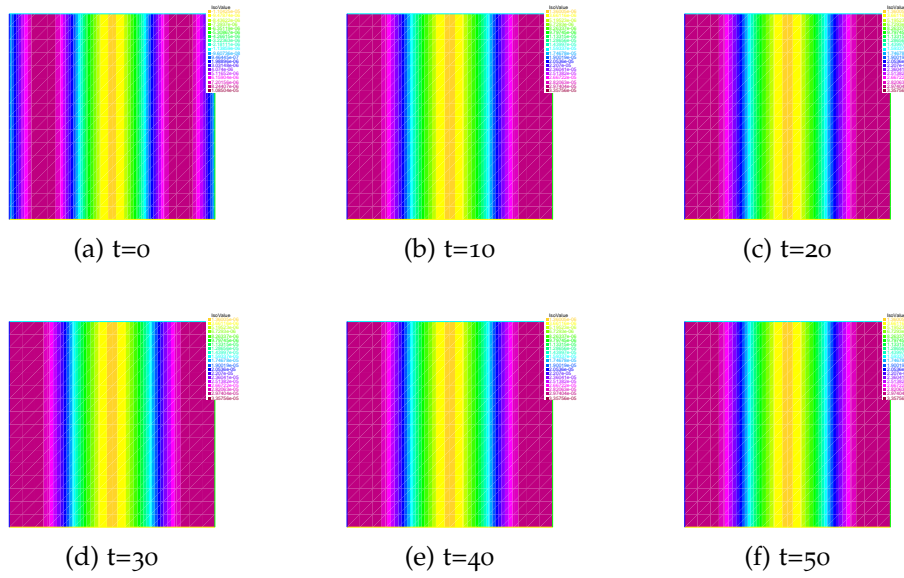


Figure 4.3: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

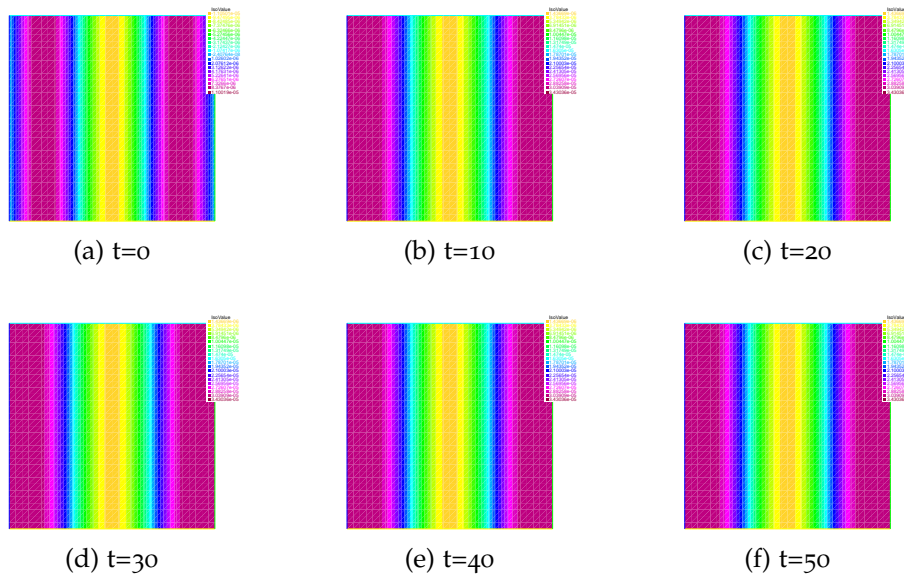


Figure 4.4: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, \pi] \times [0, \pi]$

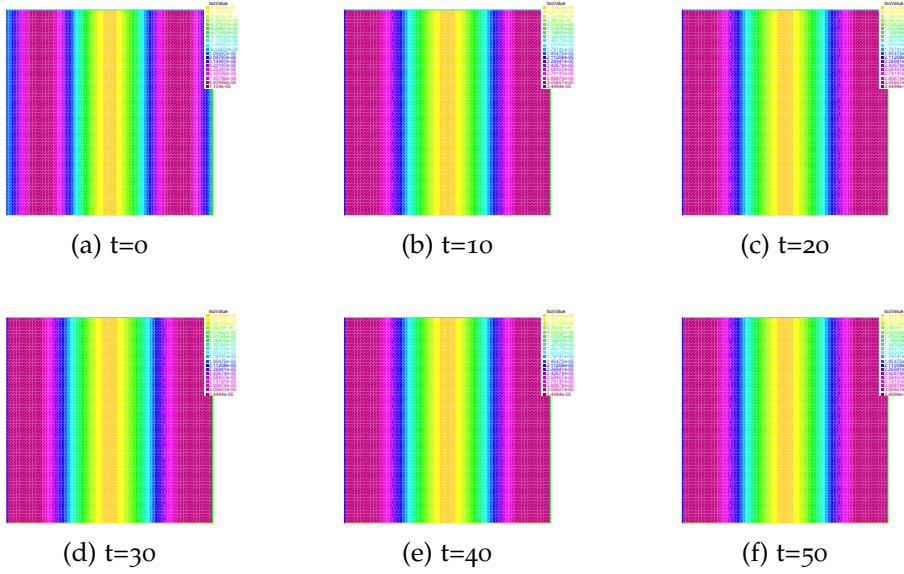


Figure 4.5: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$

4.1.2 Test 2

We set domain $\Omega = [0, \pi] \times [0, \pi]$ with $A = 12$. The initial condition $u_0(x, y) = 10^{-5} \sin(10\pi y)$ and $\tau = 0.05$, and $T = 50$. We consider the number of iterations in the x and y directions $n = 16$. We test the algorithm for the case where $n_0 = 10^{19} e^{-x}$, and $w_{ci} = 10^7$.

In Figures 4.6, 4.7, and 4.8, we plot the solutions of Algorithms 1, 2 and 3 respectively at different times, where we consider number of intervals in the x and y directions to be $n = 16$. In Figures 4.9, and 4.10, we plot the solutions of Algorithm 3 at different times whereby we consider the number of intervals in x and y directions to be $n = 32$ and 64 respectively.

The same behavior is observed for the three methods where the solution is moving in the y -direction, as shown in figures 4.6, 4.7, 4.8 for $n = 16$. Moreover, increasing n from 16, 32 to 64, i.e. decreasing h from $\frac{\pi}{16} \approx 0.19$ to $\frac{\pi}{32} \approx 0.098$ to $\frac{\pi}{64} \approx 0.049$ improves the accuracy of the

solution, while not affecting the motion in the y -direction as shown in figures 4.8, 4.9 and 4.10.

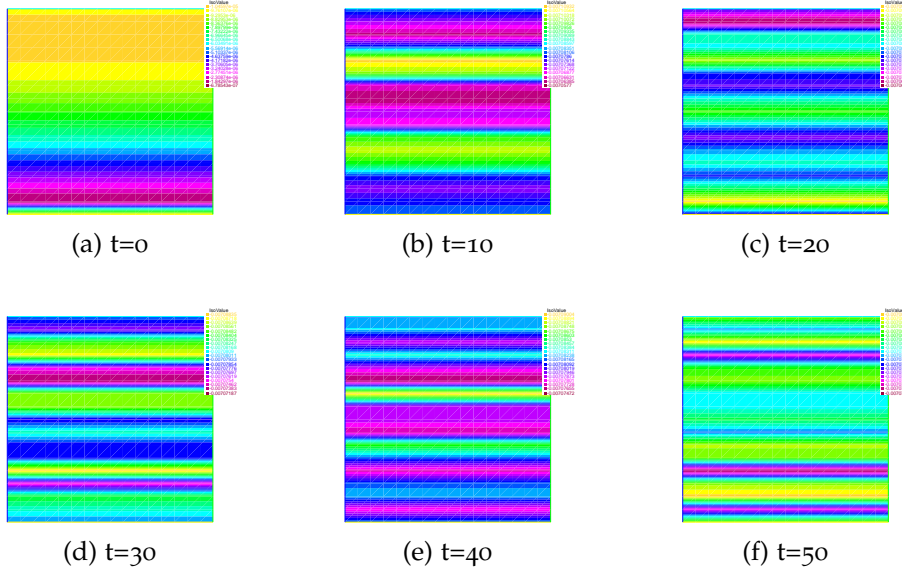


Figure 4.6: Time evolution of solution u using full newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

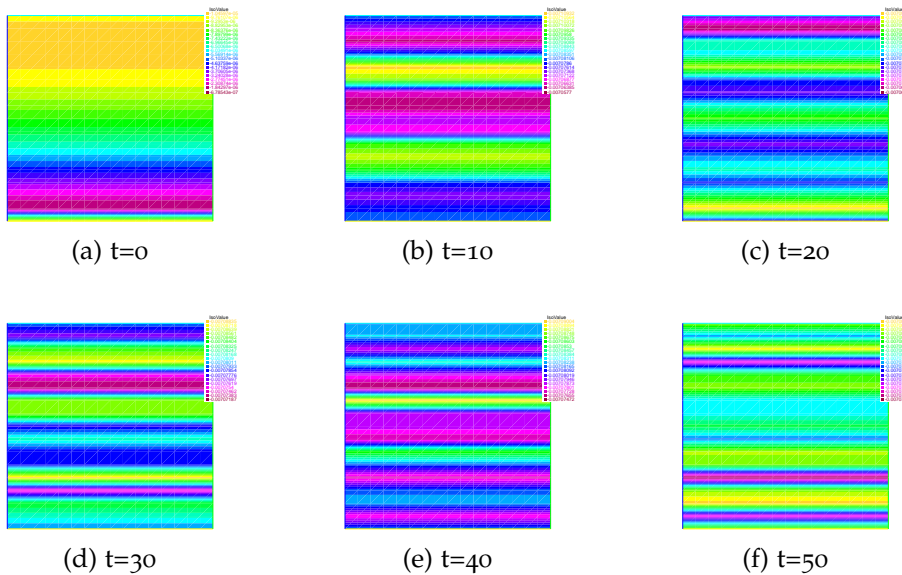


Figure 4.7: Time evolution of solution u using chord method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

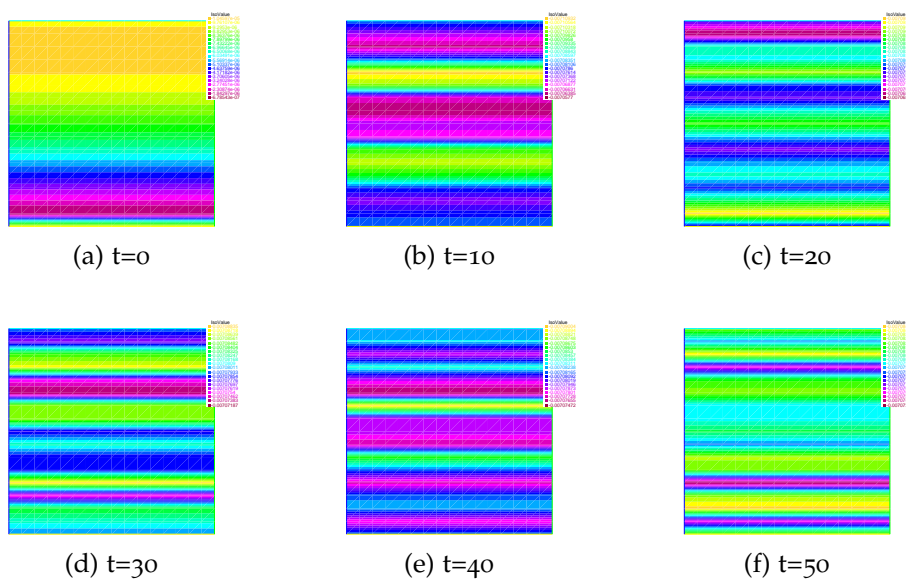


Figure 4.8: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, \pi] \times [0, \pi]$

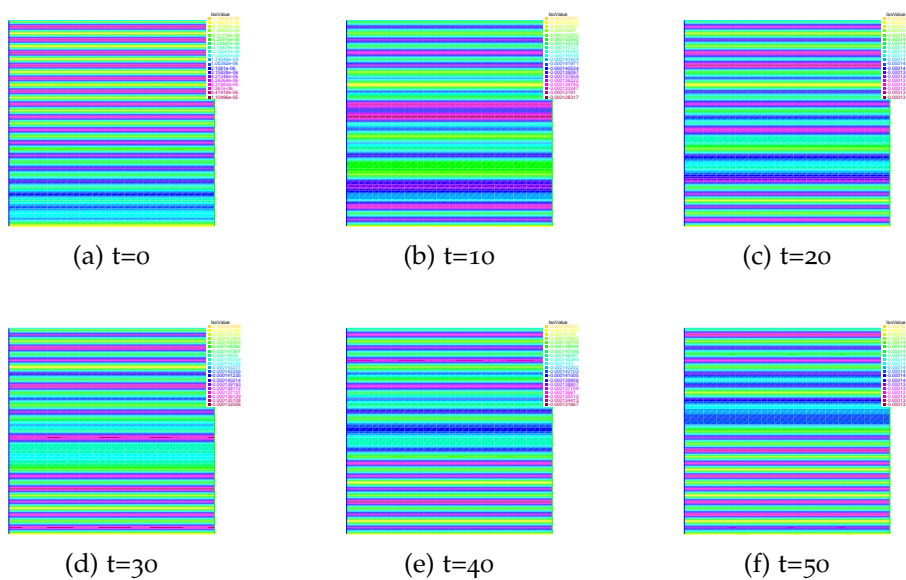


Figure 4.9: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, \pi] \times [0, \pi]$

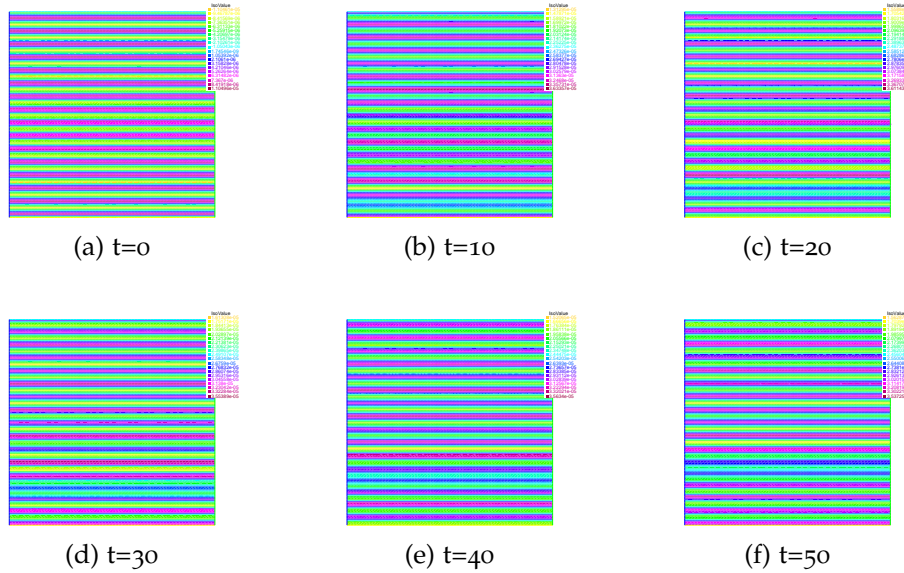


Figure 4.10: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$

4.1.3 Test 3

We set domain $\Omega = [0, 20] \times [0, 20]$ with $A = 12$. The initial condition $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$ and $\tau = 0.05$, and $T = 50$. We test the algorithm for the case where $n_0 = 10^{20}e^{-(x-10)^2/8 - (y-10)^2/8}$, and $w_{ci} = 10^7$.

In Figures 4.11, 4.12, and 4.13, we plot the solutions of Algorithms 1, 2 and 3 respectively at different times, where we consider number of intervals in the x and y directions to be $n = 16$. In Figures 4.14, and 4.15, we plot the solutions of Algorithm 3 at different times whereby we consider the number of intervals in x and y directions to be $n = 32$ and 64 respectively.

The same behavior is observed for the three methods where the solution is moving in circular motion around the center of the domain $(10, 10)$, as shown in figures 4.11, 4.12, 4.13 for $n = 16$. Moreover, increasing n from 16, 32 to 64, i.e. decreasing h from $\frac{20}{16} \approx 1.25$ to $\frac{20}{32} \approx 0.625$ to

$\frac{20}{64} \approx 0.3125$ improves the accuracy of the solution, while not affecting the circular motion as shown in figures 4.13, 4.14 and 4.15.

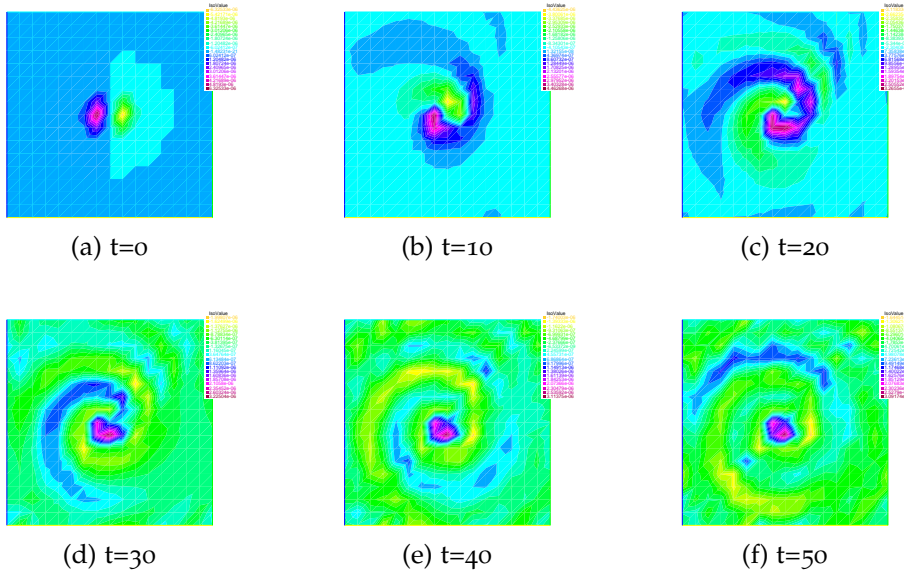


Figure 4.11: Time evolution of solution u using full newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$

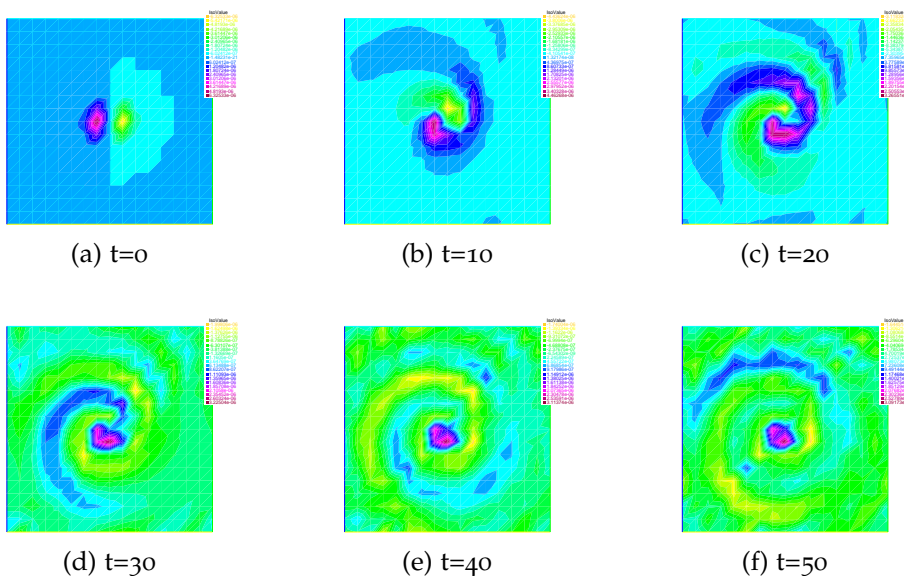


Figure 4.12: Time evolution of solution u using chord method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$

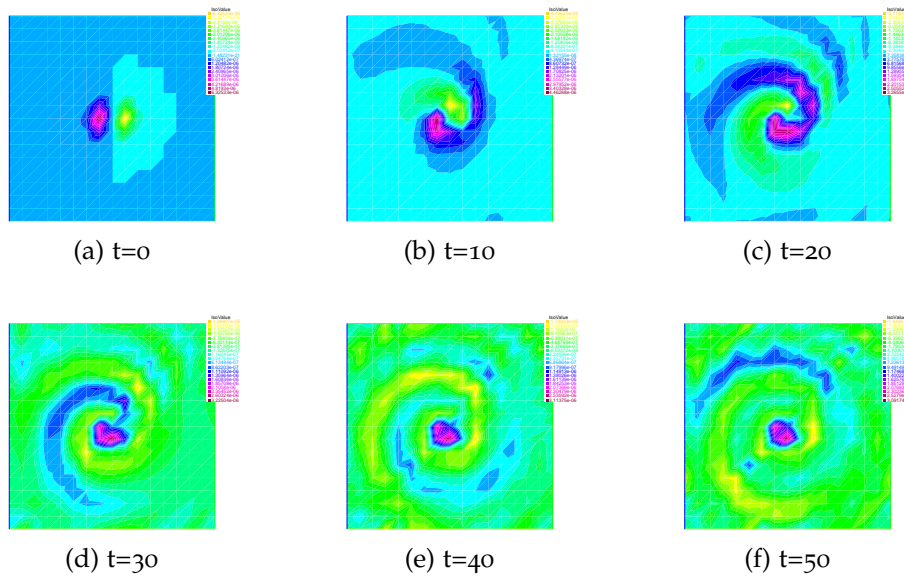


Figure 4.13: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, and a 17×17 grid on $\Omega = [0, 20] \times [0, 20]$

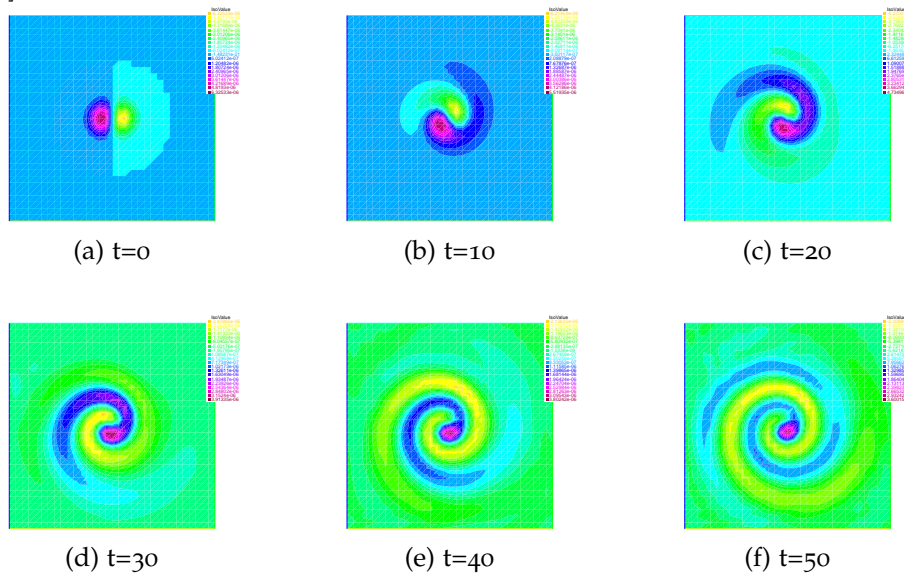


Figure 4.14: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 33×33 grid on $\Omega = [0, 20] \times [0, 20]$

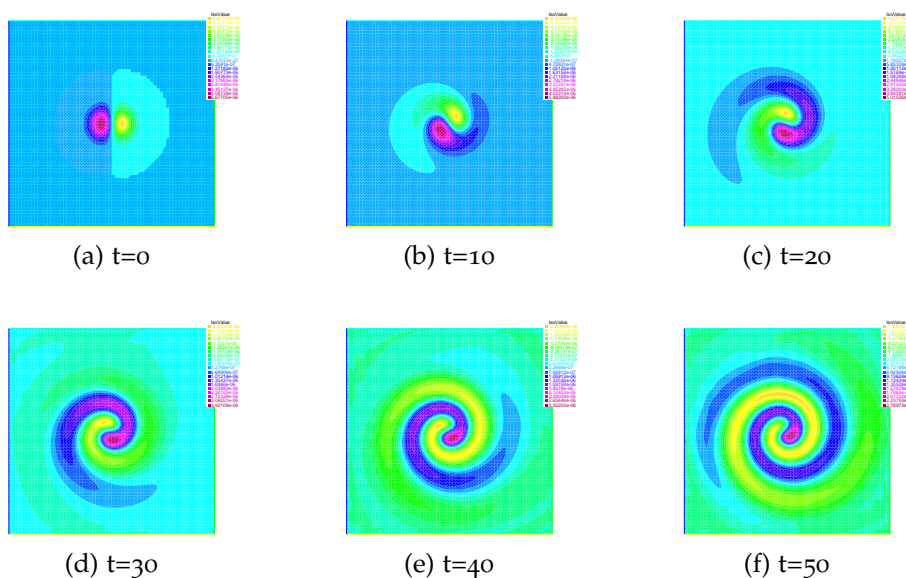


Figure 4.15: Time evolution of solution u using Modified Newton method for $T = 50$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$

4.1.4 Comparison and Analysis of results

We show the runtimes (in seconds) of the three methods for the tests with number of intervals in x and y directions, $n = 16$ in Table 4.1. Moreover, the number of iterations k per time step of the three methods for the three tests is shown in Table 4.2. Also, runtimes (in seconds) of Modified Newton's method for the three tests with different number of intervals in the x and y directions n is shown in Table 4.3. Note we consider end time to be $T = 50$.

Methods	test 1	test 2	test 3
Newton's Method	1785.1 s	3966.42 s	3691.87 s
Chord Method	2015.92 s	1591.3 sec	2158.94 s
Modified Newton Method	8.491 s	15.739 sec	17.728 s

Table 4.1: Runtimes of the three methods for three the tests number of intervals in the x and y directions, $n = 16$ and $T = 50$

Methods	test 1	test 2	test 3
Newton's Method	1	2	2
Chord Method	1	2	2
Modified Newton Method	1	2	2

Table 4.2: Number of iterations k per time step

n	test 1	test 2	test 3
16	8.491 s	15.739 s	17.728 s
32	34.706 s	67.722 s	95.765 s
64	209.37 s	542.09 s	876.72 s

Table 4.3: Runtimes of Modified Newton's methods for the three tests with different number of intervals in the x and y directions n for $T = 50$

Based on the runtimes shown in Table 4.1 for tests 2 and 3, it is clear that Modified Newton's method is the fastest, followed by Chord's method and then Newton's method.

This result is justified as matrix $B(W)$ is computed at every Newton iteration (2 times per time step). However, in chord's method $B(W)$ is computed once per time step, and in Modified Newton $B(W)$ is absent. We are unable to deduce the speed from test 1 since there is a single Newton's iteration per time step as shown in Table ???. However, according to Table 4.2, the number of iterations per time step in chord and Modified Newton is 2.

For $t = 0$, one additional iteration per time step is obtained in all tests and all methods(2 for test 1 and 3 for test 2 and test 3).

Note that as number of intervals in x and y directions increase the runtime increases as shown in Table 4.3 for Modified Newton with $n = 16$, 32 and 64.

All the tests indicate that changing the initial conditions and grid size don't affect the growth of the solution for all Newton-type methods which is not the case in semi-linear as it is shown in the next section.

4.2 Semi-linear versus Modified Newton's Method

In this section we are going to show the plots for tests 1, 2 and 3 comparing the solutions of semi-linear obtained in algorithm 1 by N-Nassif, S-Moufawad, H- Karakazian in their recent work [9] and the solutions of Modified Newton.

We set a cap of 0.3 such that for electrostatic potential values exceeding 0.3, the numerical solution will no longer be admissible and the algorithm stopped. Such cap on time was added to avoid computing non admissible physical solutions. Note that there is a $u_{max} = 0.3$ and therefore a t_{max} and if the cap not reached t_{max} by default becomes equal to end time.

We consider number of intervals in x and y directions $n = 64$

4.2.1 Test 1

In Figures 4.16, and 4.17, we plot the solutions of Algorithms 3, and semi-linear algorithm 1 in [9] respectively at different times, where we consider number of intervals in the x and y directions to be $n = 64$, and the end time $T = 200$.

In the semi-linear case, the solution remains unchanged up till $t = 31.65$, after the transition time where the solution shifts from sine function in the x -direction to a sine function in the y -direction, and the numerical solution is no longer admissible (u_{max} exceeds 0.3) before the end time $T = 200$ reached. However, in Modified Newton's method the solution remains in x -direction, unchanged and didn't exceed the admissible maximum value up till end time $T = 200$.

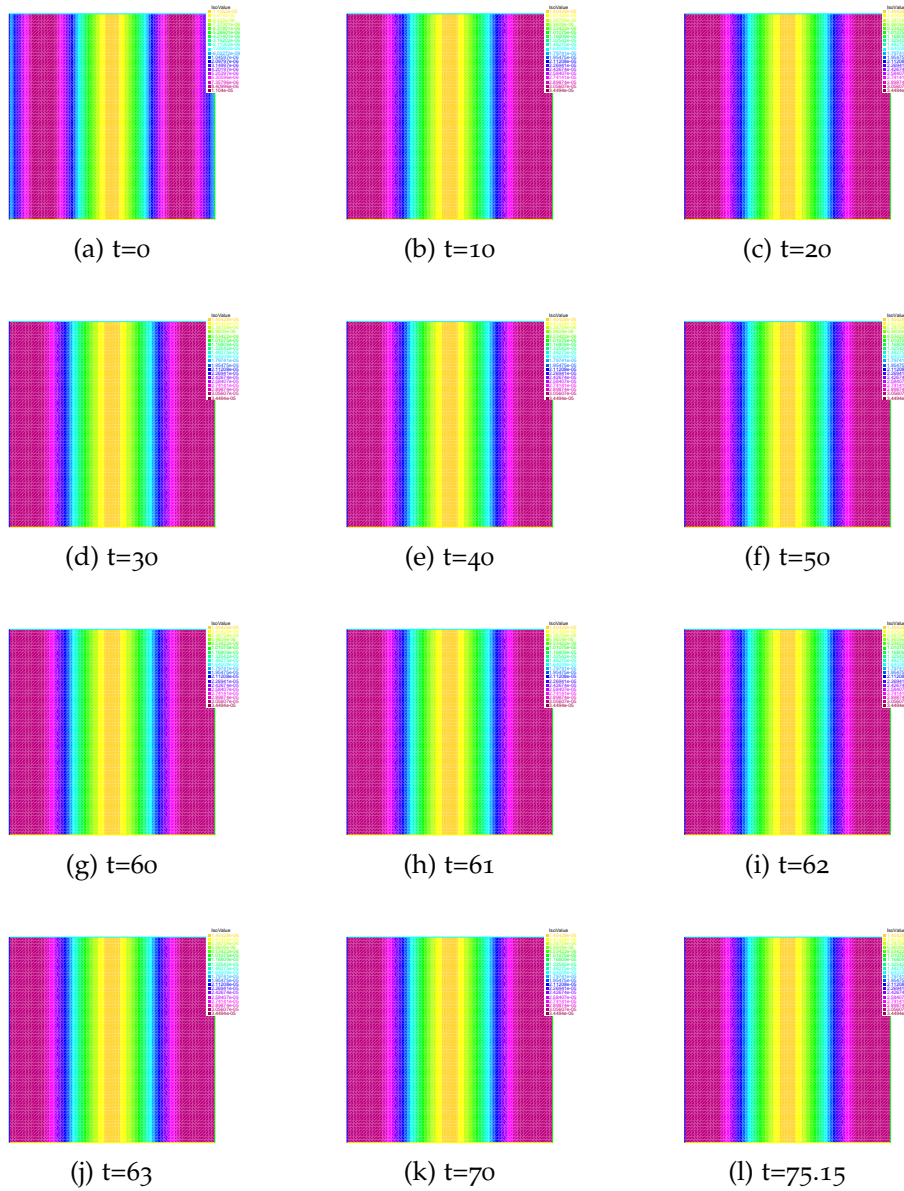


Figure 4.16: Time evolution of solution u for test 1 using Modified Newton method for $T = 200$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$

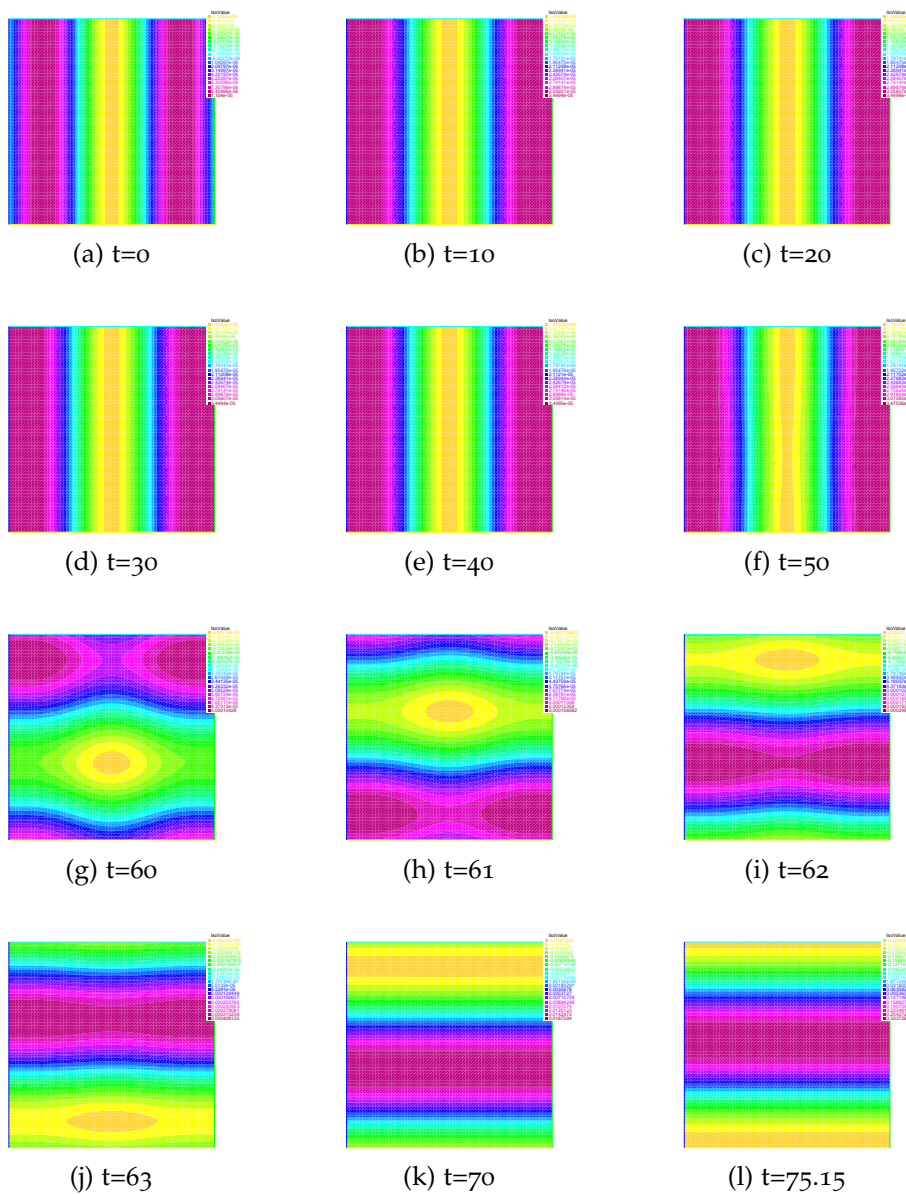


Figure 4.17: Time evolution of solution u for test 1 using semi-linear for $T = 200$, $u_0 = 10^{-5} \sin(3x)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$

4.2.2 Test 2

In Figures 4.18, and 4.19, we plot the solutions of Algorithms 3, and semi-linear algorithm 1 in [9] respectively at different times, where we

consider number of intervals in the x and y directions to be $n = 64$, and the end time $T = 200$. The solution moves in y -direction in both methods.

In the semi-linear method, u_{max} is reached very fast as the solution exceeds the admissible maximum value way before the end time $T = 200$, which is not the case for the Modified Newton's method.

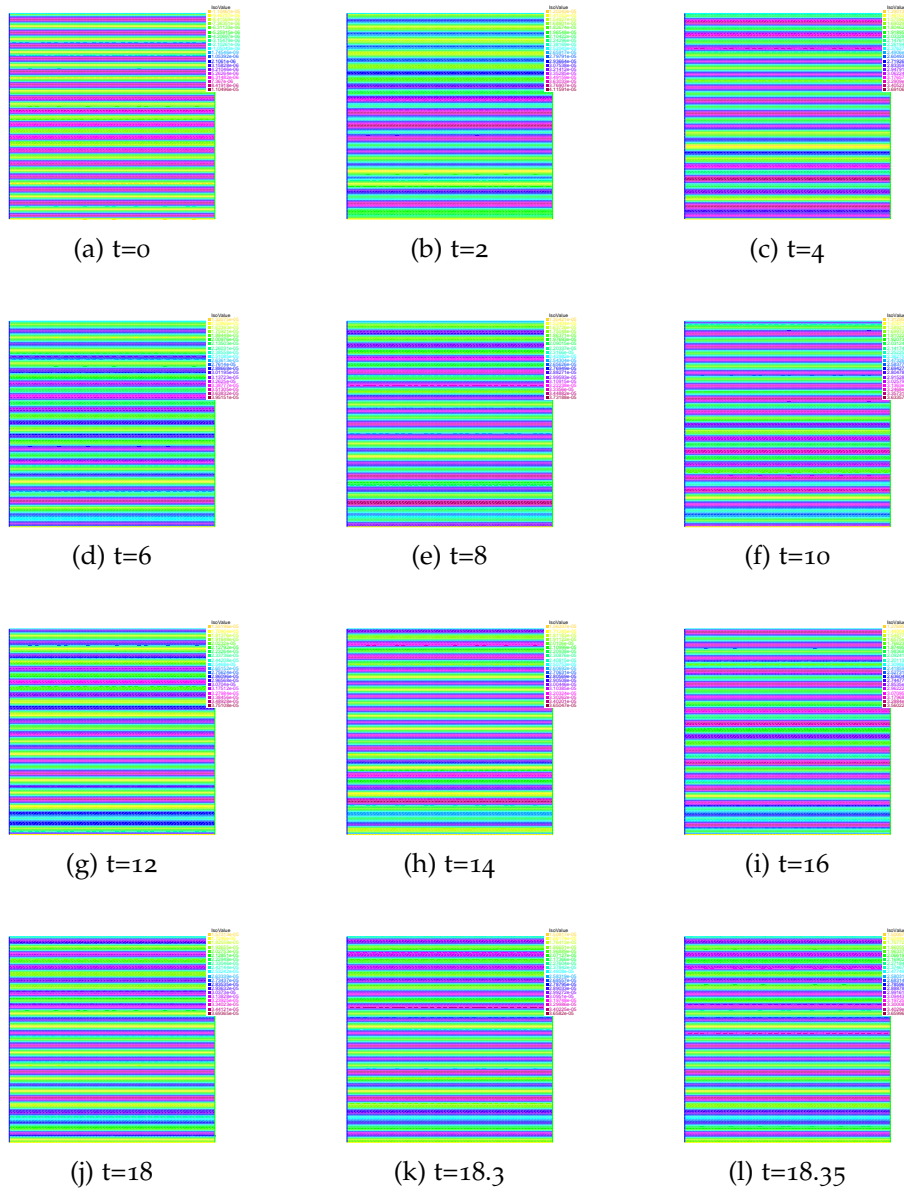


Figure 4.18: Time evolution of solution u for test 2 using Modified Newton method for $T = 200$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, \pi] \times [0, \pi]$

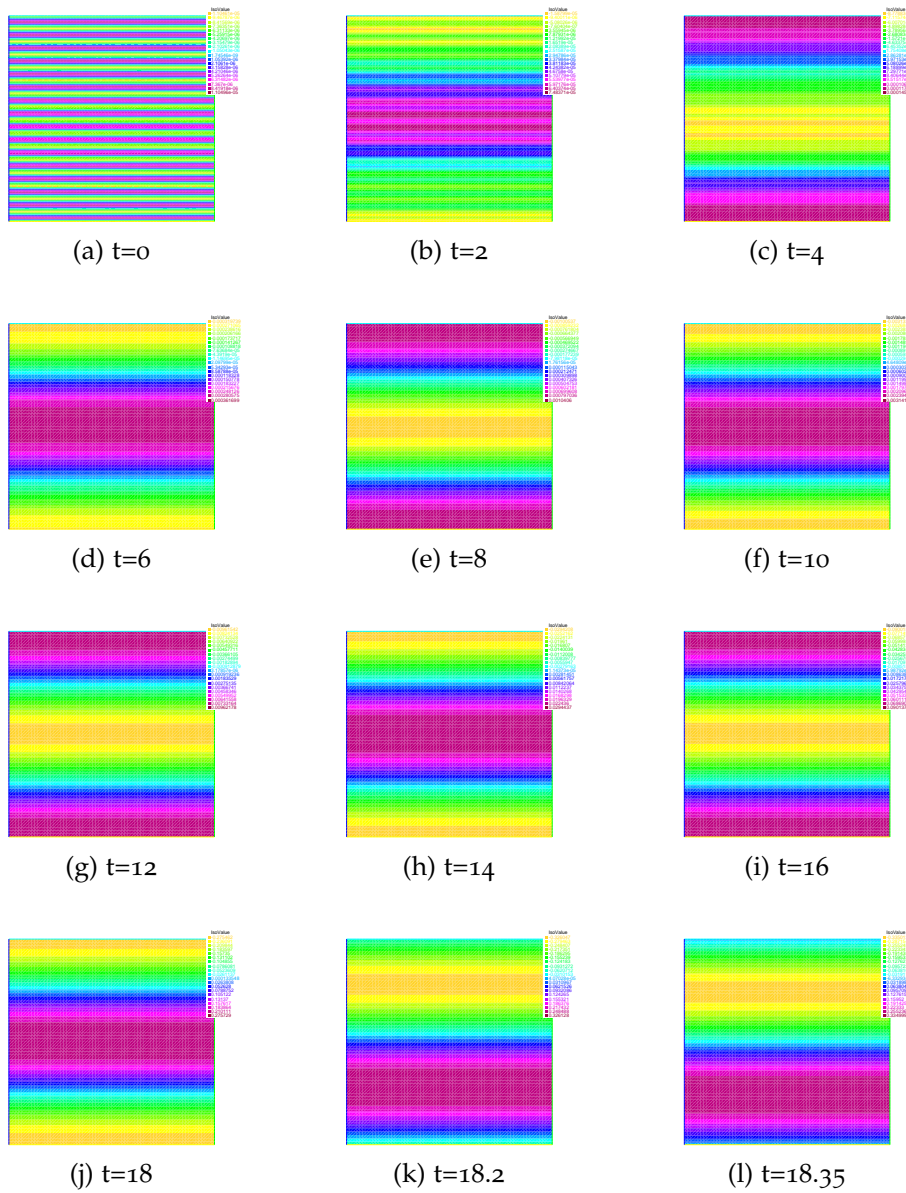


Figure 4.19: Time evolution of solution u for test 2 using semi-linear for $T = 200$, $u_0 = 10^{-5} \sin(10\pi y)$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$

4.2.3 Test 3

In Figures 4.20, and 4.21, we plot the solutions of Algorithms 3, and semi-linear algorithm 1 in [9] respectively at different times, where we consider number of intervals in the x and y directions to be $n = 64$, and the end time $T = 200$. In both methods, we have a wave moving in circular fashion around the center of the domain $(10,10)$ where the values of the solution vary slightly with respect to the initial condition.

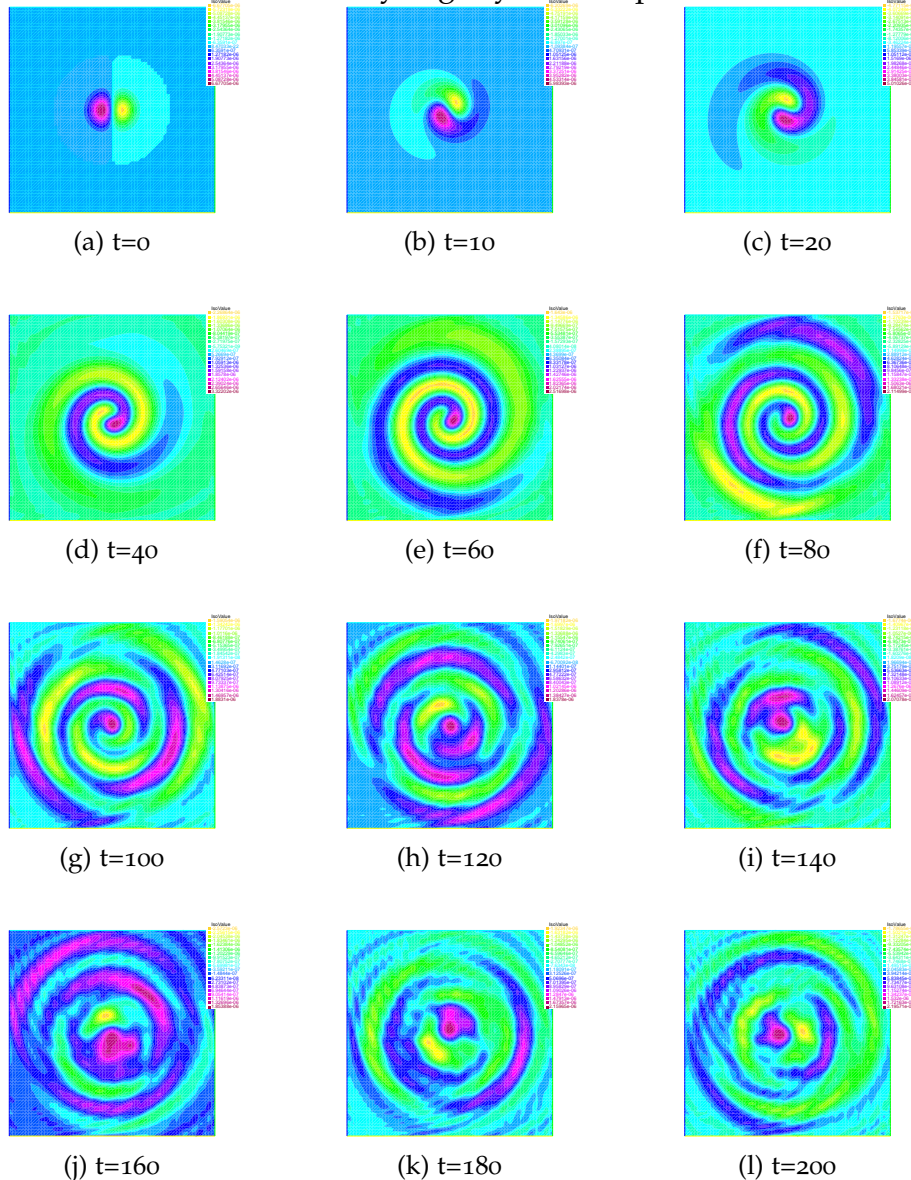


Figure 4.20: Time evolution of solution u for test 3 using Modified Newton method for $T = 200$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$

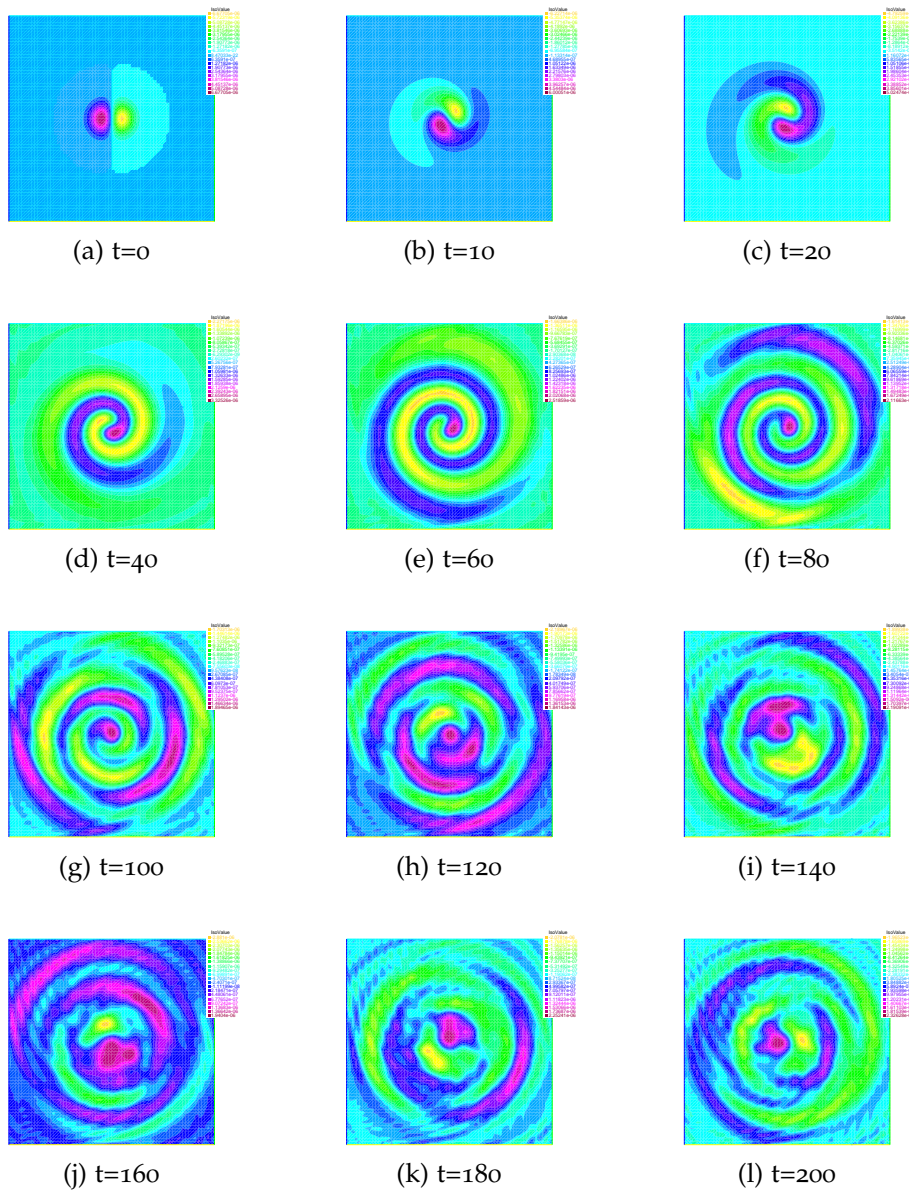


Figure 4.21: Time evolution of solution u for test 3 using semi-linear for $T = 200$, $u_0(x, y) = -10^{-5}(x - 10)e^{-0.5(x-10)^2 - 0.5(y-10)^2}$, $\tau = 0.05$, and a 65×65 grid on $\Omega = [0, 20] \times [0, 20]$

4.2.4 Comparison and Analysis of results

We show the t_{max} of the three methods for three the tests with number of intervals in x and y directions, $n = 64$ in Table 4.4. Such t_{max} corresponds to t where u_{max} exceeds 0.3 and if this cap is not reached t_{max} by default becomes equal to the end time ($T = 200$). Moreover, we show the runtimes (in seconds) of Semi-linear and Modified Newton's method for three the tests with number of intervals in x and y directions, $n = 64$ in Table 4.5. Note we consider end time to be $T = 200$.

Method	test 1	test 2	test 3
Semi-linear	75.15	18.35	200
Modified Newton	200	200	200

Table 4.4: Variation of t_{max} for semi-linear and Modified Newton's method in three tests, for $n = 64$ and $T = 200$

Method	test 3
Semi-linear	408.954 s
Modified Newton	1262.74 s

Table 4.5: Runtimes of the semi-linear and Modified Newton's method for test 3 with number of intervals in x and y directions, $n = 64$ and $T = 200$.

We put a cap of 0.3 on u_{max} to avoid computing non admissible physical solutions as for value exceeding 0.3 the numerical solution is no longer admissible. However, it should be noted that this cap is not reached in Newton's type methods for the tested cases.

According to table 4.5 and using test 3 only since $T = 200$ is reached in both methods, semi-linear is faster than Modified Newton's Method.

However, according to table 4.4, Modified Newton is numerically more stable than semi-linear version, since in all cases the solution doesn't exceeds 0.3 (unlike the semi-linear for tests 1 and 2).

Note that, In semi-linear there is a growth of the solution (test 1 and 2) unlike Newton-type methods where changing in conditions don't affect the growth of solution.

Chapter 5

Concluding Remarks

In this thesis, we have handled the Euler implicit nonlinear fully discrete system (3.1.2) using Newton's type methods: Full Newton's method (Algorithm 1), Chord's method (Algorithm 2) and Modified Newton's method (Algorithm: 3).

Full Newton's Method is slower than Chord's Method due to the computation of the matrix $B(W)$ several times per time step as opposed to once per time step. Moreover, it should be noted that the newly introduced Modified Newton's method (Algorithm 3) avoids the computation of the matrix $B(W)$, thus outperforms all the other Newton's type methods in terms of runtime. In addition, the replacement of $B(W)$ by an approximation did not affect the robustness of the algorithm, since very similar (if not the same) solutions as the Full Newton and Chord's methods were obtained.

Newton's type methods, including Modified Newton's method, are more costly than explicit and semi-linear schemes. This is due to the necessity of one or more iterations per time step which requires updating the corresponding Jacobians, thus leading to an increase in the processing time.

However, this cost is worth investing in, given that the resulting simulations are more representative of the non-linear phenomenon due to the Poisson's bracket $\{u, \Delta u\}$.

As a matter of fact, Newton-type simulations were carried out without putting a cap on the electrostatic potential u . This was not the case for semi-linear discretization used in [9]. Such point is extremely meaning-

ful in building up a robust algorithm for the Hasegawa-Mima system.

Theoretical aspects regarding the tested algorithms are not tackled in this thesis. For completion, future work may consider treating those aspects of the problem.

Bibliography

- [1] D. Almaalouf. A new simulation model for the Hasegawa-Mima plasma equation. PhD thesis, 2017.
- [2] F. Hariri. Simulating Bi-dimensional turbulent plasmas using the Hasegawa-Mima model-by Farah Assadallah Hariri. PhD thesis, 2010.
- [3] A. Hasegawa and K. Mima. Stationary spectrum of strong turbulence in magnetized nonuniform plasma. Phys. Rev. Lett., 39:205–208, Jul 1977.
- [4] A. Hasegawa and K. Mima. Pseudo-three-dimensional turbulence in magnetized nonuniform plasma. Physics of Fluids - PHYS FLUIDS, 21:87–92, 01 1978.
- [5] A. Hasegawa and M. Wakatani. Plasma edge turbulence. Phys. Rev. Lett., 50:682–686, Feb 1983.
- [6] Frédéric Hecht. New development in freefem++. Journal of numerical mathematics, 20(3-4):251–266, 2012.
- [7] W. Horton. Statistical properties and correlation functions for drift waves. The Physics of fluids, 29(5):1491–1503, 1986.
- [8] W. Horton. Drift wave vortices and anomalous transport. Physics of Fluids B: Plasma Physics, 1(3):524–537, 1989.
- [9] H. Karakazian, S. Moufawad, and N. Nassif. A finite-element model for the hasegawa-mima wave equation, to appear jcam.
- [10] V. Larichev and G. Reznik. Two-dimensional solitary rossby waves. Doklady Akademii nauk SSSR, 231:1077–1079, 01 1976.
- [11] J. Nycander. The effect of the electron temperature gradient on nonlinear drift waves in plasmas. Physica Scripta, 39(6):758, 1989.
- [12] S. Strintzi. Simple models of plasma turbulence.
- [13] P. Terry and W. Horton. Stochasticity and the random phase approximation for three electron drift waves. Physics of Fluids - PHYS FLUIDS, 25:491–501, 03 1982.

- [14] P.W. Terry and W. Horton. Drift wave turbulence in a low-order k space. Phys. Fluids; (United States), 26:1, 01 1983.
- [15] RE Waltz. Numerical study of drift wave turbulence with simple models for wave-wave nonlinear coupling. The Physics of Fluids, 26(1):169–179, 1983.