Fourier Spectral Methods for Numerical Modeling of Ionospheric Processes

by

Atikah Ismail

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

APPROVED:

[Signatures]

Dr. W. A. Scales, Chairman

[Signatures]

Dr. I. M. Besieris

[Signatures]

Dr. R. O. Claus

May 1994
Blacksburg, VA
Fourier Spectral Methods for Numerical Modeling of Ionospheric Processes

by

Atikah Ismail

Chairman: Dr. Wayne A. Scales

Electrical Engineering Department

(ABSTRACT)

Fourier spectral and pseudospectral methods are used in numerical modeling of ionospheric processes, namely macroscopic evolution of naturally and artificially created ionospheric density irregularities. The simulation model consists of two-dimensional electrostatic nonlinear fluid plasma equations that describe the plasma evolution. The spectral and pseudospectral methods are used to solve the spatial dependence of these self-consistent equations. They are chosen over the widely used finite difference and finite element techniques since spectral methods are straightforward to implement on nonlinear equations. They are at least as accurate as finite difference simulations. A potential equation solver is developed to solve the nonlinear potential equation iteratively. Time integration is accomplished using a combination of leapfrog and leapfrog-trapezoidal methods. A FORTRAN program is developed to implement the simulation model. All calculations are performed in the Fourier domain.

The simulation model is tested by considering three types of problems. This is accomplished by specifying an initial density (Pedersen conductivity) profile that represents slab model density, density enhancement (due to releases such as barium), or density depletion (due to late times effects of electron attachment material releases) in the presence of a neutral wind. The evolution of the irregularities is monitored and discussed. The simulation results agree with similar results obtained using finite difference methods. A comparison is made between the ionospheric depletion and enhancement problems. Our results show that, given the same parameters and perturbation level, the depletion profiles bifurcate much faster than that of the enhancement. We argue that this is due to the larger growth rate in the $\mathbf{E} \times \mathbf{B}$ interchange instability of the density depletion case.
Acknowledgements

First, I would like to express my gratitude and sincere appreciation to my advisor Dr. Wayne A. Scales. I am especially grateful for his enthusiasm and patience in helping me learn the basics of space plasma physics and in solving the many problems I encountered during my research. He has provided extremely helpful and valuable advice. Without his support and encouragement, this thesis would never be accomplished.

I would like to thank Dr. Ioannis M. Besieris and Dr. Richard O. Claus for serving in my committee and for their valuable guidance I received throughout my undergraduate and graduate studies here at Virginia Tech.

I would like to thank Glenda Scales for all her help in videotaping the animation of my simulation runs.

I would like to thank the members of the Satellite Communications group and many others for their friendship. I would like to express my appreciation to Kelly, Marcia and Michael Carter (and Sailor) for making me feel like a part of their family.

Finally, I would like to thank my parents Andar and Constance Ismail and my brother Sjarif for their continuing love, encouragement and support. This work is dedicated to you.

This work was supported by Office of Naval Research (ONR) grants N00014-92-J-1498 and N00014-92-J-1484.
List of figures

Figure 2.1: Atmospheric nomenclature based on temperature, mixing and ionization .... 4
Figure 2.2: Typical variation of free electron density with height measured from
    ground level ................................................................. 5
Figure 2.3: Variation of effective electron collision frequency with height .......... 6
Figure 2.4: Attenuation of ionization in the ionosphere ........................................ 8
Figure 2.5: Atmospheric composition to 1000 km for a typical temperature profile ... 11
Figure 2.6: Suggested schematic description of the evolution of a barium ion cloud
    in a plane perpendicular to the magnetic field .................................. 22
Figure 2.7: Development of striation from images taken on Ba release over Eglin
    Air Force Base in 1971 ................................................................ 23
Figure 2.8: Snow plow effect following the passage of large rocket .................... 25
Figure 2.9: Large-scale ionospheric plasma depletion caused by the launch of
    Skylab in 1973 ..................................................................... 26

Figure 4.1: 2-D simulation box set-up ................................................................. 47
Figure 4.2: Physical picture of a fluid element in the 2-D fluid simulation ............ 42
Figure 4.3: Accuracy comparison of passive scalar convection of a cone .............. 50
Figure 4.4: Attenuation filter for various values of kmax and n ............................. 60
Figure 4.5: Leapfrog integration method ............................................................. 62
Figure 4.6: Leapfrong-trapezoidal integration method .......................................... 63

Figure 5.1: Physical mechanism of the E × B instability .................................... 70
Figure 5.2: Basic algorithm of ESSPEC2 ............................................................. 73
Figure 5.3: Initial density n (conical distribution) on 128 × 128 grid space .......... 77
Figure 5.4: Velocity field for uniform rotation ...................................................... 78
Figure 5.5: Contours of density n obtained after $\frac{1}{4}$ revolution ......................... 78
Figure 5.6: Density n (conical distribution) after 1 revolution on 128 × 128
    grid space ......................................................................... 80
Figure 5.7: Initial density n (gaussian distribution) on 128 × 128 grid space ........ 81
Figure 5.8: Density n (gaussian distribution) after 1 revolution on 128 × 128
    grid space ....................................................................... 82
Figure 5.9: Velocity field for translation in the positive x-direction ...................... 83
Figure 5.10: Density n (gaussian distribution) of translation in the positive
    x-direction ......................................................................... 84-85
Figure 5.11: Density contours for $\rho(x, y) = \cos \left(\frac{2\pi}{L} x\right) \sin \left(\frac{2\pi}{L} y\right)$ ............................. 87
Figure 5.12: $\phi = \frac{1}{13} \left( \frac{L}{2\pi} \right)^2 \cos 2 \left( \frac{2\pi}{L} x \right) \sin 3 \left( \frac{2\pi}{L} y \right)$ as solved analytically from $\rho$ ........... 88
Figure 5.13: $\phi$ as computed by subroutine POTENTIAL ........................................... 88
Figure 5.14: Laplacian of $\phi$, should be identical to $-\rho$ ........................................ 89
Figure 5.15: Slab model of plasma ................................................................. 91
Figure 5.16: Initial density profile of the slab model of plasma .................................. 91
Figure 5.17: Late times density profile of the slab model of plasma ......................... 92
Figure 5.18: Isodensity contours for the plasma cloud enhancement evolution ........ 95-96
Figure 5.19: Surface plots of the initial and late-time density profiles of the density enhancement problem ............................................. 97
Figure 5.20: Energy history plot for density enhancement evolution ....................... 98
Figure 5.21: Isodensity contours for the plasma cloud depletion evolution ............. 101-102
Figure 5.22: Surface plot of the late-time density profiles of the density depletion problem ......................................................... 103
Figure 5.23: Energy history plot for density depletion evolution .......................... 103
Figure 5.24: Comparison of density profiles at $T = 30$ of the enhancement and depletion models ....................................................... 104
Figure 5.25: Comparison of equipotential profiles at $T = 30$ of the enhancement and depletion models ............................................. 106
Figure 5.26: Comparison of density profiles at late-times ($T = 60$) of the enhancement and depletion models ..................................... 107
# List of tables

Table 2.1: Classification of layers in the ionosphere ........................................ 4

Table 5.1: Typical parameter for ionospheric plasma 200 - 300 km .................. 72
# Table of contents

Abstract ............................................................. ii  
Acknowledgement .................................................. iii  
List of figures ....................................................... iv  
List of tables ........................................................ vi  
Table of contents ................................................... vii

Chapter 1: Introduction and objectives ........................ 1  
Chapter 2: The ionosphere and its irregularities .......... 3  
  2.1. Ionospheric formation ........................................ 7  
  2.2. Physical parameters of the ionosphere ................... 10  
  2.3. Natural irregularities in the ionosphere ................. 13  
    2.3.1. Scintillations ........................................... 13  
    2.3.2. Irregularities at equatorial regions ................. 14  
    2.3.3. Travelling ionospheric disturbances ................. 15  
    2.3.4. High latitude irregularities ............................ 15  
    2.3.5. Effects of ionospheric irregularities on communication systems ... 16  
  2.4. Ionospheric modification ................................... 16  
    2.4.1. Ionospheric chemical releases ........................ 17  
    2.4.2. Chemical process in ionospheric chemical releases .... 18  
    2.4.3. Plasma cloud injection methods ........................ 20  
    2.4.4. Observation and detection of ionospheric chemical releases .......... 26  
    2.4.5. Barium release ........................................... 21  
    2.4.5. Electron attachment material release ................ 24  

Chapter 3: Basic plasma principles ............................. 27  
  3.1. Debye shielding .............................................. 27  
  3.2. Plasma parameter ............................................ 29  
  3.3. Plasma frequency ............................................ 29
3.4. Charged particles gyromotion in uniform $B$ fields .................................. 31
3.5. Kinetic description of plasma ......................................................... 33
3.6. Fluid description of plasma .............................................................. 36
  3.6.1. Derivation of fluid equations .................................................. 36
  3.6.2. Complete set of equations for fluid description in plasma .......... 38
  3.6.3. Incompressible fluid ............................................................... 39
3.7. Electrostatic approximation of ionospheric plasma ......................... 40
  3.7.1. The concept of $\beta$ .............................................................. 40
  3.7.2. Electrostatic Maxwell's equations ......................................... 41
  3.7.3. Motion parallel and perpendicular to $B$ .................................... 41

Chapter 4: Numerical simulation methods ............................................. 43
  4.1. Classification of numerical simulation of plasmas ....................... 43
    4.1.1. Fluid simulation method .................................................... 44
    4.1.2. Particle simulation ............................................................ 44
    4.1.3. Vlasov and Fokker-Planck codes ....................................... 45
    4.1.4. Hybrid codes ................................................................. 46
  4.2 Two-dimensional electrostatic fluid simulation ............................... 46
  4.3. Spectral method in spatial discretization .................................... 49
    4.3.1. General spectral methods .................................................. 51
    4.3.2. Spectral methods using Fourier series .................................. 52
    4.3.3. Differentiation ............................................................... 53
    4.3.4. Convolution ..................................................................... 55
    4.3.5. Comparison of spectral and pseudospectral approximation ....... 59
  4.4. Time integration ....................................................................... 59
  4.5. Numerical accuracy requirements ................................................. 65

Chapter 5: Numerical simulation of some cases of ionospheric irregularities ......................................................... 66
  5.1 Theoretical model ...................................................................... 66
    5.1.1. Theoretical model in real space .......................................... 67
    5.1.2. Theoretical model in spectral domain .................................... 70
    5.1.3. Energy conservation ............................................................ 71
  5.2 Two-dimensional simulation model ................................................. 72
    5.2.1. Fast Fourier Transform ....................................................... 73
1 Introduction and objectives

The evolution of artificial plasma clouds due to chemical releases in the earth’s ionosphere has been under scientific investigation for more than three decades. A variety of experimental programs have been carried out, beginning with sodium trail releases in 1955 [Davis, 1979]. Recent experiments include the barium releases of CRRES (Combined Release and Radiation Effects Satellite) mission and the releases of electron attachment materials of NICARE (Nickel Carbonyl Release Experiment) project. These releases create plasma density enhancements and depletions at late times. Along with experimental efforts, the study of the artificial plasma cloud evolution has been pursued theoretically [Perkins et al., 1973; Huba et al., 1988] and numerically using computer modeling [Zabusky et al., 1973; Zalesak et al., 1985; Scales and Bernhardt, 1991] to understand the physical and dynamical processes associated with artificial and natural ionospheric irregularities.

In this thesis work we are interested in numerical modeling the ionospheric processes due chemical releases using a simple, idealized cloud models. This problem was originally addressed by Zabusky et al. in 1973 and has since been modified by other plasma physicists to include relevant ionospheric effects. Hence, this is a very well-studied problem. However, to our knowledge, no studies or simulation have been performed using the numerical methods that we plan to use in our work. Most, if not all, simulation work of this type of problem employ a variety of finite difference or finite element methods. We propose to develop and solve a two-dimensional electrostatic, nonlinear and self-consistent fluid model that describes the plasma evolution, using Fourier spectral and pseudospectral techniques.

Fourier spectral and pseudospectral methods have distinct advantage over finite difference or finite element methods. First, nonlinear terms can be easily handled by the spectral methods. In contrast, solving nonlinear equations using finite-difference techniques is much more complicated and cumbersome, at best. Second, for solving continuous and periodic functions, spectral methods are at least as accurate as other methods. Furthermore, spectral methods have a substantial speed advantage over other numerical methods.

It is the goal of this research work to develop a two-dimensional simulation model that describes the plasma cloud evolution. Fourier spectral and pseudospectral methods
will be used in decomposing the spatial dependence of the associated partial differential equations dictated by our model. Our work will provide the groundwork for more realistic 3-D simulations using spectral methods in future investigations.

The organization of this thesis is as follows. In chapter 2 we briefly discuss the formation and important parameters of the ionosphere. Natural irregularities and their effects on communication systems will be described. We then consider the ionospheric modifications through chemical releases which are related to the upcoming topic on numerical simulation on plasma evolution. Chapter 3 provides some of the basic principles of plasma theory, with emphasis on fluid behavior of plasmas. Techniques used in numerical simulation are the central topic of chapter 4. An overview of different types of plasma simulations will be presented along with the descriptions of the physical set-up of our simulation. The majority of this chapter is devoted to detailed discussion on spectral method techniques as used in solving fluid plasma equations. We then implement the techniques described in chapter 4 to study important ionospheric processes in chapter 5. Artificial ionospheric irregularities produced by chemical releases will be simulated. We present and discuss our simulation results and then compare them to the results from experimental observations and other numerical simulation of comparable nature. Finally, we will summarize our results in chapter 6.
2 The ionosphere and its irregularities

The Earth’s atmosphere can be described by four properties: pressure, temperature, density and composition. These properties determine the atmosphere’s behavior. Based on the variation of these properties, the vertical structure of the atmosphere is commonly classified as follows:

- Temperature regimes (troposphere, stratosphere, mesosphere, thermosphere)
- Chemical composition regimes (homosphere, heterosphere)
- Ionization regimes (ionosphere, magnetosphere)
- Dynamics/Mixing regimes (barosphere, exosphere)

The classification of the atmosphere’s structure is given in Figure 2.1. In this thesis, we are mostly interested in the electrically charged portion of the atmosphere, known as the ionosphere. The ionosphere is characterized by a permanent population of ions and free electrons due to low pressure that inhibits quick recombination in the ionized species. The charged particles greatly influence the medium’s electrical properties, even though their density may amount to only 1% of the neutral particle density.

The existence of a conducting region in the higher levels of the terrestrial atmosphere was speculated by William Thompson (Lord Kelvin) and was investigated in 1882 by Balfour Stewart in conjunction with daily magnetic variations [Tascione, 1988]. However, it was not until Guglielmo Marconi’s well-known experiments of transmitting a radio signal from Cornwall, England to Newfoundland, Canada in 1901, did the studies of this phenomena become widespread. Arthur Kennelly and Oliver Heaviside in 1902 independently postulated that, because of the Earth’s curvature, the radio waves must have been reflected from an ionized layer. Appleton confirmed the existence of such region two decades later. The word ionosphere was coined by R. Watson-Watt in 1926. Since then the ionosphere has become very important for communications in the high frequency range 3 - 30 MHz.

The vertical structure of the ionosphere changes continuously. It varies from day to night, with the seasons of the year and with latitude. Figure 2.2 shows the different iono-
Figure 2.1: Atmospheric nomenclature based on temperature, mixing and ionization. (After Hargreaves [1992].)

Table 2.1: Classification of layers in the ionosphere.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Altitude</th>
<th>Electron density</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>60-90 km</td>
<td>$10^8 - 10^{10} \text{m}^{-3}$</td>
</tr>
<tr>
<td>E</td>
<td>105-160 km</td>
<td>several $10^{11} \text{m}^{-3}$</td>
</tr>
<tr>
<td>F1</td>
<td>168-180 km</td>
<td>several $10^{11} - 10^{12} \text{m}^{-3}$</td>
</tr>
<tr>
<td>F2</td>
<td>maximum variable around 300 km</td>
<td>up to several $10^{12} \text{m}^{-3}$</td>
</tr>
</tbody>
</table>

spheric layers. In the order of increasing altitude and increasing electron concentration, these layers are called D, E, F1, and F2. Note that beyond the F peak, the electron density decreases exponentially with increasing altitude. The characteristics of the ionospheric layers are summarized in Table 2.1 [Hargreaves, 1992].

Figure 2.2 also shows the typical daytime and night time vertical electron density profiles. During the daytime the F layer splits into layers F1 and F2. Not only does the overall electron density in the lower ionosphere decrease at night, the D region vanishes soon after sunset while the E region becomes much weaker. The sunset and sunrise effects, however, are almost nonexistent in the F region.
Figure 2.2: Typical variation of free electron density with height measured from ground level. (After Griffiths [1987].)
Figure 2.3: Variation of effective electron collision frequency with height. (After Griffiths [1987].)

The E and F layers are the most important for radio communications in the frequency range of 3 to 30 MHz. Above 40 MHz, the radio waves will penetrate through the ionosphere. The range of usable frequencies in radio communications depends largely on the variation of the ionosphere's collision frequency with height, as shown in Figure 2.3. Only frequencies smaller than the collision frequency can be reflected. In the D region the collision frequency is high enough so that only the lower frequency waves (2 MHz or below) can be reflected, the higher frequency waves are absorbed.

This chapter will deal with several of the physical aspects of the ionosphere, particularly the ionospheric formation and the composition. Since the ionosphere is an integral part of radio communications in that the presence of irregularities in the ionosphere may affect our communication systems, we will discuss various types of natural irregularities. Methods of producing ionospheric modification with artificial irregularities as means to studying the natural irregularities will also be presented in this chapter.
2.1 Ionospheric Formation

The ionosphere is principally formed by the photoionization of atmospheric gases such as $\text{N}_2$, $\text{O}_2$ and $\text{O}$. The extreme ultra-violet (EUV) and X-ray parts of the spectrum of solar radiation incident on the upper atmosphere is absorbed at various altitudes and by various gaseous entities before reaching the lower atmosphere. As depicted in Figure 2.4, the ionizing energy encounters an increasing concentration of ionizable atmospheric particles as it penetrates the geospace. It will then produce ionization at an increasing rate. However, since the photons are absorbed in the process of phonoionization, the solar beam itself decreases in intensity as it penetrates, causing an offset in the atmospheric concentration. A peak rate of ion production will be attained, while at lower heights the rate will decline until the flux of ionizing energy becomes negligible.

Once formed, the electrons and ions tend to recombine and to react with other gaseous species to produce other ions. Depending on the relative speed of the production and loss processes, there is a dynamic equilibrium of the net concentration of free electrons. This process can be expressed by a continuity equation:

$$\frac{\partial N}{\partial t} = q - L - \nabla \cdot (Nv), \quad (2.1)$$

where $N$ is the electron density, $q$ the production rate, $L$ the loss rate by recombination, and $\nabla \cdot (Nv)$ expresses the loss of electrons by movement with $v$ the mean drift velocity.

The first term on the right hand side of (2.1), the rate of ion-electron pairs production, can be expressed as

$$q = \eta \sigma n I, \quad (2.2)$$

where $I$ is the intensity of ionizing radiation at some level of the atmosphere and $n$ is the concentration of available atoms or molecules for ionization by radiation. The amount of radiation absorbed by an ionized atom or molecules is expressed by the absorption cross section $\sigma$. The ionization efficiency $\eta$ is the fraction of the absorbed radiation that goes into producing ionization.

Sydney Chapman first treated the problem of simple ionospheric layer formation and its daily variation in his 1931 paper. The Chapman production function describes the rate of production of ionization $q$. Hargreaves [1992] points out some assumptions made in the
Figure 2.4: Production of ionization in the atmosphere. (After Belrose [1965].)

derivation:

1. The atmosphere is of single species composition, exponentially distributed with constant scale height;

2. There are no variations in the horizontal plane of the atmosphere;

3. Solar radiation is absorbed proportionally following (2.2);

4. The radiation is monochromatic.

An in depth derivation of the Chapman production function can be found in Tascione [1988] or Hargreaves [1992].

The Chapman production function is commonly written in the normalized form:

\[ q = q_{m0} e^{1-z-sec \chi} e^{-z} \]  \hspace{1cm} (2.3)

where \( z = (h - h_{m0})/H \) is the reduced height for the neutral gas, \( H \) is the scale height, \( \chi \) is the solar zenith angle, and \( h_{m0} \) is the height of maximum production rate when the Sun is overhead (\( \chi = 0 \)). By differentiating (2.3) with respect to \( z \), we find that the maximum production rate \( q_m \) in general is equal to \( \eta l_{\infty} / eH \) sec \( \chi \) at \( z_m = \ln(\sec \chi) \).
Electrons produced in the upper atmosphere tend to reunite with positive ions to reform neutral particles and to attach themselves to neutral molecules to form negative ions such as $\text{O}_2^-$. This recombinant and attachment is represented by the second term of the continuity equation (2.1). Assuming that no negative ions is present and the electrons recombine directly with the positive ion, the recombinant rate is $L = \alpha N_e^2$ where $N_e$ is the electron density and $\alpha$ is the recombinant coefficient. If the electron drift is neglected, $\zeta = \alpha N_e^2$ at equilibrium. Then from the Chapman production function, we obtain

$$N_e = N_{e0} e^{\frac{1}{2} \left(1-z - \sec \chi x e^{-z} \right)}$$  \hspace{1cm} (2.4)

Maximum electron density occurs at $N_m = N_{e0} \sqrt{\cos \chi}$. A layer with these properties is called an $\alpha$-Chapman layer.

Next, we assume that electrons are removed by becoming attached to neutral molecules. The rate of electron loss is $L = \beta N_e$ where $\beta$ is the attachment coefficient. $\beta$ is a function of altitude since it depends on the concentration of neutral molecules. The rate is linear with $N_e$ because the neutral species is numerous compared to the number of free electrons. At equilibrium $q = \beta N_e$ and from the Chapman production function,

$$N_e = N_{e0} e^{1-z - \sec \chi x e^{-z}}$$  \hspace{1cm} (2.5)

Maximum electron density varies as $N_m = N_{e0} \cos \chi$. This layer is called $\beta$-Chapman layer.

Much of the recombinant in F and F1 regions are through dissociative recombinant reactions. This is a two-stage process: first, the positive ion combines with one of the common molecular species; second, the resulting charged molecule is dissociated through recombinant with an electron. The F2 region, while possessing some recombinant, cannot be explained using a Chapman model because neither $\alpha$ (recombination) nor $\beta$ (attachment) dominates. A simple Chapman model cannot be applied to the D region either. The D region is the most complex part of the ionosphere from the chemical point of view due to the relatively high pressure and significant contribution to ion production by several different sources of ionization. Readers interested in an in depth derivation of regions D and F2 formation are referred to Harmanes [1992] and Tascione [1988].

Finally, we consider the last term of the right-hand side of the continuity equation (2.1). This term accounts for changes of electron concentration caused by the movement of
ionization. Assuming that photochemical production and loss are negligible in comparison with the effects of movement, from the continuity equation we obtain

$$\frac{\partial N}{\partial t} = -\nabla \cdot (N \mathbf{v}),$$  \hspace{1cm} (2.6)

where $\mathbf{v}$ is the drift velocity of the plasma. Furthermore, if we are only interested in the vertical movement and supposing that the drift is due to diffusion, we have

$$\frac{dN}{dt} = \frac{\partial}{\partial h} \{D_p \left( \frac{dN}{dh} + \frac{N}{H_p} \right) \},$$  \hspace{1cm} (2.7)

where $h$ is the height, $D_p$ is the plasma diffusion coefficient, and $H_p$ is the plasma scale height. At equilibrium, $dN/dH = -N/H_p$ with the plasma exponentially distributed as $N/N_0 = \exp(-h/H_p)$ with scale height $H_p$. This distribution has the same form as a Chapman layer but with about twice the scale height. As photochemistry at greater heights becomes less important, diffusion will dominate. A complete derivation can be found in *Hargreaves* [1992].

The preceding paragraphs discuss the dynamic equilibrium of electron density through production, recombination and vertical transport in which the production is based on photoionization. There are other sources of ionization in the ionosphere. Ionization by energetic particle impact on the neutral gas is particularly important at high latitudes. Visible light is also emitted when particles strike the atmosphere. These light emissions create the visible aurora.

2.2 Physical parameters of the ionosphere

Before we go into the ionospheric irregularities, it is worth while to review some of the physical parameters of the ionosphere. We place the emphasis of our discussion on the E and F regions, the regions of interest. Figure 2.5 depicts the atmospheric chemical composition for a typical temperature profile. A comprehensive graph of electron density profile and temperature profile versus height has been shown previously in Figure 2.1. We can draw some valuable insights from these figures.

Due to radiative cooling, the temperature reaches minimum in the range of 130 - 250°K in the D region. $N_2$ and $O_2$ remain the dominant constituents at this altitude. The per-
Figure 2.5: Atmospheric composition to 1000 km for a typical temperature profile. (After Hargreaves [1992].)
entage of ionization is exceedingly low and the properties of the neutral atmosphere have a significant influence on the ionization. The specific heat of the atmosphere at this layer is so large that even though the rate of heat input almost vanishes at night, the temperature changes very little. The temperature in the E and F layers is increasingly variable, both diurnally and with solar activity. It increases rapidly above 120 km until it reaches 1500 °K at 300 km (in the F region). This temperature increase can partially be explained by absorption of another portion of the solar UV spectrum. The thermal velocity of electrons in the F region is several hundred km/s.

The E region peaking at 105 - 110 km is formed by the more penetrating parts of the spectrum, compared to those in the F region. EUV radiation is absorbed by molecular oxygen to form \( O_2^+ \). The primary ions are \( N_2^+, O_2^+ \) and \( O^+ \), but the most numerous ones are observed to be \( NO^+ \) and \( O_2^+ \) [Hargreaves, 1992].

The lower F region is attributed to the most heavily absorbed part of the solar radiation. The principal constituents are atomic oxygen and dominating molecular nitrogen. As the altitude increases, the dominant constituent changes from atomic nitrogen to atomic oxygen. Above the peak of the F layer, the molecular forms are seen to be of minor importance. Atomic oxygen which previously predominates is gradually replaced by helium. Eventually hydrogen becomes the major species above 3000 km.

Overall reactions in both E and F regions are controlled by direct dissociative recombination of the type

\[
e + XY^+ \rightarrow X + Y.
\] (2.8)

Dissociative recombination is \( 10^5 \) times faster than the direct radiative recombination. The electron and ion loss proceeds via molecular ions, where \( X \) and \( Y \) in (2.8) are replaced by \( N \) and \( O \) in the E layer main recombinations. In the F region the principal primary ion is \( O^+ \), which is first converted to a molecular ion by a charge exchange reaction followed by the dissociative recombination that leaves \( NO^+ \) and \( O_2^+ \) as the most abundant products.

The recombination rate in the upper part of the F2 region is the \( \beta \) type where \( \beta \) depends on the concentration of \( N_2 \). On the other hand, the production rate depends on the \( O \) concentration. Since the ratio of the masses of \( N_2 \) and \( O \) is 1.75:1, the loss rate falls off more quickly than the production rate. This layer, the Bradbury layer, is characterized
by the increase of the electron density with increasing heights. The increasing electron density is balanced by the diffusion in the F2 layer that produces decreasing electron density distribution as the height increases. Where these two processes are equally important, at about 250 to 500 km, the maximum ionospheric electron density occurs. This is called the F peak. Since the F2 region has the greatest concentration of electrons, it is also the region of greatest interest in radio propagation. Unfortunately, it is also the most variable, the most anomalous and the most difficult to predict.

2.3 Natural irregularities in the ionosphere

The previous sections lay down the basic explanation of the formation and parameters of the ionosphere. The ionosphere, to a large extent, varies in a regular and predictable manner. However, irregularities and perturbations such as storms, solar flares and other anomalies and peculiarities do occur from time to time. These perturbations are usually large in distance and slow to change in time. Ionospheric irregularities, on the other hand, do not last long and are small enough to produce spatial variations within a field of a single observing instrument. Due to their effects on communication systems, ionospheric irregularities have been widely studied using various techniques such as radar, ionosonde, beacon satellite, airglow photometer, and in-situ measurement from satellite or rocket [Hargreaves, 1992].

We begin this section with some of the natural irregularities present in the low and mid latitude ionosphere followed by irregularities found in the upper latitude. Finally, we will review some of the effects that ionospheric irregularities have on radio propagation and satellite communications.

2.3.1 Scintillations

Scintillation arises when the irregularity of the medium produces phase irregularity along the wavefront which will then be converted to amplitude variations by the process of diffraction. Most scintillation occurs in the F region. At mid latitude, F region scintillations tend to occur in patches about 1000 km across. These patches have been identified with the phenomenon of Spread-F. F region irregularities also show considerable elongation along
the direction of the geomagnetic field. This is due to the field-aligned plasma diffusion in the F region.

2.3.2 Irregularities at equatorial regions

Equatorial scintillation is closely related to the range-spread type of Spread-F which is attributed to irregularities of electron density. Early observations using ionosondes showed that the reflected echo did not display a well-behaved pattern but was "spread" in range or frequency. Irregularities of the equatorial F region have been thoroughly investigated because of the severe effects they have on radio propagation. Measurements using rockets and satellites have shown large reductions of plasma density in the equatorial F region by night. The equatorial Spread-F (ESF) is caused by the ability of gravity waves to organize the equatorial plasma into high and low density regions with the same horizontal wavelength as the gravity waves. Once this perturbation occurs, the generalized Rayleigh-Taylor instability (GRT)* can take over and causes oscillation to grow [Kelley, 1989].

Irregularities in the equatorial E region, present both day and night, are strongly correlated with the equatorial electrojet. The electrojet which is more intense over the magnetic equator is part of the global current system generated by the tidal motions of the Earth's atmosphere. Since the east-west drift velocity of the electrons exceeds the ion-acoustic speed, the two-stream instability may develop, generating irregularities in the ionized medium. This type of irregularities gives echoes characterized by a narrow spectrum with a Doppler shift that corresponds approximately to the ion acoustic velocity in the electrojet region. They are known as 'Type 1 echoes' or two-stream irregularities†. The second type of equatorial irregularity, 'Type 2 echoes', is explained by the gradient drift instability‡. The spectral width of this type of irregularity is much broader than that associated with the

---

*Steep upward-directed gradient develops in the bottomside of the nighttime F layer. The gravity forces downward. Lower density plasma is advected upward in the depletions region, creating a larger perturbation, and the system is unstable.

†Streams of electrons and ions differ in velocity by more than the ion-acoustic speed gives rise to electrostatic waves. Oscillation energy is gained and the plasma becomes unstable.

‡Specific type of GRT instability when the effects of gravity is neglected. May also be called 'E x B drift' instability. The electric current provides the relative motion of electrons and ions across the magnetic field which allows instability to develop. A more detailed explanation will be discussed in chapter 5.
Type 1 echoes. It is observed that Type 2 echoes disappear after sunset if the daytime electrojet reverses direction from eastward to westward [Hargreaves, 1992].

2.3.3 Travelling ionospheric disturbances

Travelling ionospheric disturbances (TID) are produced by auroral heating events that send out high-altitude atmospheric gravity waves. These disturbances travel from high latitudes toward the equator almost vertically. Near the active auroral regions, large upwellings of heated gas from the lower atmosphere drastically change the composition of the neutral and ions at higher altitudes. An average TID can produce about 1% variation in the electron content, small but readily detectable. Possible TID sources include earthquakes, atmospheric tides, turbulence in thermospheric winds, and large scale weather systems.

2.3.4 High latitude irregularities

Large regions of enhanced ionization are observed in the polar cap and the auroral regions. These enhancements have high density which is more typical of the daylight mid-latitude ionosphere. They are usually formed some distance away but drift in the polar convection to the polar cap. In the auroral zone some of the enhancements peak in the E region or the lower F region.

Scintillations at high latitude tend to be particularly severe around the auroral zone and the polar cap. The occurrence and intensity maximizes at night. Similar to scintillations at mid and low latitudes, the irregularities are strongly field-aligned such that their effects are also enhanced along the local magnetic field direction. It has been observed that small-scale irregularities which produce scintillations were located at the edges of large-scale enhancements [Kelley, 1989]. Mechanisms such as the gradient drift and Kelvin-Helmholtz\footnote{Consider a layer of fluid flows over a stationary layer. If there is a small perturbation in the flow, an outward centrifugal force will appear at each bend in the perturbation. The bend will thereby reinforced and the instability will grow.} instabilities can cause large-scale enhancement to break up at the edges, hence generating smaller enhanced patches.

The high latitude irregularities of the E region are associated with electric currents from the auroral electrojet. As in the equatorial region of the middle and lower latitudes, this phenomena gives rise to Type 1 and Type 2 echoes that are attributed to the two-stream
and gradient drift mechanisms, respectively. A notable difference exists in that the deduced velocity may far exceed the ion acoustic speed in the auroral zone but not in the equatorial zone.

2.3.5 Effects of ionospheric irregularities on communication systems

In general, natural irregularities degrade a communication channel by inducing amplitude or phase fluctuations as a result of scattering or multipath mixing. The effectiveness of the ionosphere in radio propagation depends on its variability due to perturbations and irregularities. Scintillations, Spread-F and travelling disturbances affect the performance of HF communication circuits by focusing or defocusing the signal, causing fading, or allowing propagation over two or more paths simultaneously with consequent interference. TIDs tend to degrade the accuracy of HF direction finders used in search and rescue operations.

The most serious ionospheric effect in communications involving links between satellites and ground stations is scintillation. The presence of scintillation requires an increase in the transmitted power and thereby increasing the cost, to maintain adequate signal to noise ratio. Scintillation effects due to the ionosphere invariably fall as the radio frequency increases. However, scintillations may still be a problem in the equatorial regions during periods of high solar activity.

2.4 Ionospheric modification

We have seen how the HF communication systems can be negatively affected by the natural irregularities in the ionosphere. Not surprisingly, studies on ionospheric irregularities have been widely performed since 1920s. Most upper atmospheric studies are conducted by observing natural phenomena. However, since 1955, various materials have been injected into the high atmosphere for the purpose of creating observable tracers or perturbations to the ambient medium and ongoing processes within it. Optical techniques have been used to observe the effects of ionospheric releases, along with other methods such as radio frequency, magnetic and in situ particle counting. By introducing known perturbations into the ionosphere under tightly controlled conditions, it has become feasible to study how the imposed perturbation interacts with the ionosphere and how the integrity of the communication
systems may be affected. Many active experiments have been conducted at mid latitude to model instabilities caused by either gravity or by large-scale secondary perturbation of electric field. Aside from generating artificial plasma density structures, these experiments performed at the F region can be used to control existing irregularities or study processes governing the evolution of the irregularities.

Active experiments techniques in the ionosphere include the use of space shuttles and sophisticated rocket payloads, multi-satellite experiments and observations, and ground-based radiowave injection. The modification experiments include:

- particle beam injection
- chemical releases
- ionospheric heating
- wave injection

Together with computer simulation work, modification experiments provide invaluable advancement in ionospheric studies.

Although each of the above modification techniques merits our interest, we restrict the scope of our work to the ionospheric chemical releases. In this section we shall discuss the chemical processes involved in chemical releases in the F region at altitude 200 - 400 km along with the methods for release, observation and detection. Different examples of chemical releases and their effects on the ionosphere will be presented.

2.4.1 Ionospheric chemical releases

Artificial ionospheric irregularities can be produced by injecting neutral substances into the upper atmosphere. A chemical release produces a localized charge in plasma density which can lead to enhanced transport in chemically modified regions. The generation of localized density perturbation can be explained by the electron and ion continuity equation. Chemical releases can directly affect the production rate \( q \) and the loss rate \( L \) in (2.1). It enhances the production through photo or collisional ionization of neutral vapors. It also accelerates losses through electron attachment or dissociative recombination processes. In short, chemical releases produce ionospheric modification in the F region by:
• chemical enhancement of the electron density

• chemical reduction of the electron density

• physical convection of the plasma from one region to another

Ionospheric modification by chemical release excites artificial enhancements in airglow intensities by exothermic chemical reactions between the newly created plasma species. Immediately following a neutral gas release, the dense neutral cloud couples itself to the background plasma. This phenomena is appropriately called coupling and it occurs through the modification of the transport term \( (\nabla \cdot N \mathbf{v}) \) in the continuity equations. The motion of the plasma following a chemical release is commonly considered in terms of parallel and perpendicular transport relative to \( \mathbf{B} \). Due to the relatively large parallel conductivity, the parallel and ambipolar electric fields are much less than polarization fields across the magnetic fields.

2.4.2 Chemical process in ionospheric chemical releases

There are four classes of initial chemical reactions in the upper atmosphere following a neutral gas release [Bernhardt and Scales, 1990]. The first class is oxidation. This reaction is of the form

\[
XY + O \rightarrow XO^* + Y,
\]  

(2.9)

where \( XY \) represents the injected substance and \( XO^* \) is the product of the reaction which may be left in an excited state. Atomic oxygen can react rapidly with the released material, hence reducing the efficiency for ionospheric modification. Due to the reduction of modification efficiency and the airglow production that contaminates the emissions used to diagnose plasma chemistry, the Class I neutral reactions are generally unwanted.

The second class of reaction of ionospheric modification is photoionization. This reaction is generally represented by

\[
XY + h\nu \rightarrow XY^+ + e^-.
\]  

(2.10)

Photoionization yields enhanced ion and electron concentration from sunlight exposure on easily ionizable materials such as barium, cesium, sodium, and europium. Class II reactions enable optical observation of the release if the parent neutral and/or the product
ion fluoresce in sunlight. The main drawback of this type of reactions is that sunlight is required to produce the modification.

The third class of reaction in the upper atmospheric modification is the positive-ion, molecule charge transfer. The form of this reaction is

$$XY + O^+ \rightarrow XO^+ + X,$$

where $XO^+$ is a positive molecular ion which can rapidly recombine with electron. Reaction in (2.11) is followed by the dissociative recombination reaction,

$$XO^+ + e^- \rightarrow X^+ + O^*,$$

where the * indicates excited states. An example of this reaction is the CO$_2$ release at thermal speed. Class III reactions eventually lead electron density reductions after the molecular ions (XO$^+$) have recombined. The primary disadvantage of this class of reactions is that it requires a two step process to produce reduction in electron density.

The fourth class of reaction directly depletes the electron concentration by dissociative electron attachment using

$$XY + e^- \rightarrow X^- + Y,$$

where $X^-$ is the negative ion dissociation product. The initial electron attachment reactions are very fast compared to the initial ion-molecule reactions (as in class III reactions). Reaction rates of the electron attachment reactions depend on the temperature and decrease rapidly below the activation energy. The negative ion dissociation product either reacts with the atomic oxygen ion

$$X^- + O^+ \rightarrow X^* + O^*,$$

or becomes photo-dissociated

$$X^- + h\nu \rightarrow X + e^-.$$

The main limitation for this type of ionospheric modification is that they must be conducted in darkness for long-lived effects. Class IV reactions are preferred for experiments which need the maximum changes in longitudinal conductivity.
2.4.3 Plasma cloud injection methods

There are two methods of creating plasma clouds. We shall use barium as an example of the injected chemical. The first method is to fill a canister with barium (Ba) grains and copper oxide (CuO) powder. An ignition starts a chemical reaction between Ba and CuO when the rocket is at the correct altitude. The heat of the reaction bursts the canister and the excess barium is expelled and vaporized.

The second method uses high explosives to shoot a jet of barium gas in a selected direction. This is called the shaped charge technique. Using this technique, the jet may trail for more than 10 km/s and can reach the heights of tens of thousands kilometers.

2.4.4 Observation and detection of ionospheric chemical releases

Observation of the chemical trail and its deformation by air motion of materials that ionize and exhibit strong resonance scattering in sunlight such as barium, cesium, lithium, and strontium is most generally accomplished by optical techniques. The measurement method is essentially one of triangulation. It requires a minimum of two observing stations strategically located. Each station has to have a sufficiently dark sky to allow detection of the traces against the background, and yet that the chemical trail receives enough sunlight to fluoresce and scatter. The optical techniques most commonly used involve cameras and some form of photometric device.

Neutral and ion winds are measured by optically recording the motion of luminous clouds that are produced by chemicals released from rockets during twilight. The motion of these luminous trails represent the ion drift velocity vector. The small amount of un-ionized strontium atoms contained in the Ba chemical releases resonantly scatter solar radiation. Their motion gives the velocity of the prevailing neutral wind.

Aside from observation of visible modification effects, the studies of the ionized upper atmosphere is mostly interested in obtaining the electrons and ions concentrations, temperatures, and drift velocity of either the ambient thermal electrons or thermal ions. A mass spectrometer is required to determine the composition, but information on concentrations and temperatures can be obtained from simpler devices such as probes, traps, and analyzers. These measurements are commonly done in situ using instruments mounted on satellites and rockets that are moving through the plasma at velocities between 1 and 9 km/s. A
conducting surface will collect electron and ion currents that can be calculated by assuming that the plasma has a drifting Maxwellian distribution function. If the conductor is held at some potential $P$, the current is calculated by integrating the distribution function over the collector surface area for all energies greater than $P$. Additional electrodes are often incorporated to allow a more detailed analysis to be made in real time. The results are transmitted to a ground station by telemetry.

The Langmuir probe is used to measure electron temperature. It consists of a small conducting surface with cylindrical or spherical geometries. Retarding Potential Analyzer determines the temperature and density of ions by measuring the ambient ion current to a collector as a function of an applied retarding potential. Similar to a Langmuir probe operation, a curve of ion current versus retarding potential is obtained and the thermal ion temperature is determined. Ion drift velocity is measured by an Ion Drift Meter which gives components of the drift velocity normal to the motion of the satellite or rocket. Mass spectrometers for ion composition measurement is utilized for the detection of very low concentrations of constituent species. Other devices are incapable of detecting ions if the conducting surface is too small or if the number of ions striking the conductor is too small. Most mass spectrometers employ high-sensitivity detectors called electron multipliers to detect the presence of a single charged particle [Kelley, 1989].

2.4.5 Barium release

The barium release in the ionospheric modification experiments have been well studied for the last thirty years. Over 100 barium releases have been performed at altitude from 150 to 60,000 km. Barium is the most frequently used tracer species because its short photoionization constant of about 30 seconds and since excited barium ions emit visible light and are therefore easily observed. If the tracer aspect is important, the height-integrated conductivity of the barium clouds must be less than that of the ionosphere. Barium release is of the Class II reactions which creates ionospheric density enhancement.

Mapping the electric and magnetic fields of barium vapor cloud release at altitude above 150 km was first described by Lüst and his colleagues at the Max-Planck Institute [Haerendel and Lüst, 1970; Haerendel et al., 1967]. Almost all barium clouds are observed to deform and striate after having moved across the magnetic field a distance equal to only several
Figure 2.6: Suggested schematic description of the evolution of a barium ion cloud in a plane perpendicular to the magnetic field. (After Zalesak et al. [1985].)

Times their initial diameter [Rosenberg, 1971].

Spatially irregular density that grows with time, known as striation, develops in both small and large barium ion clouds. The development of striation is explained in Davis [1979]. Depending upon the magnitude of the ambient electric field, subsequent to the release of barium mixture, ion and neutral clouds move in separate directions. The ion cloud usually becomes elongated in the direction of motion. The portion most distant from the neutral cloud retains a stable configuration. However, the portion closest to the neutral cloud develops a steepened profile as shown in Figure 2.6. It then evolves into sheets 200-1000 m thick, aligned with the magnetic field. Subsequently, the individual sheets develop into rod-like structures with typical diameter of 200-400 m and spaced 700-1000 m apart. Figure 2.7 shows the development of striation based on the images taken from barium release made at 194 km altitude over Eglin Air Force Base, Florida in 1967.
Figure 2.7: Photographs and density traces of developing Ba cloud viewed up the magnetic field lines. (After Rosenberg [1971].)
The formation of striations has been pursued theoretically by various scientists including Perkins et al. [1973], Simons et al. [1980], and Haerendel et al. [1967]. Numerical simulation work on the evolution of artificial plasma clouds created by neutral barium release has been widely performed also [Zabusky, 1973; Zalesak et al., 1983, Huba et al., 1988, Mitchell et al., 1985, and many others].

2.4.6 Electron attachment material release

Recently, release of chemicals that directly attach electrons and produce electron depletions and heavy negative ion clouds have received renewed interest. Chemicals used in the recent experiments, CRRES (Combined Release and Radiation Effects Satellite) and NICARE (Nickel Carbonyl Release Experiment), include sulfur hexafluoride (SF₆), trifloro methyl bromide (CF₃Br) and nickel carbonyl (Ni(CO)₄) [Scales and Bernhardt, 1991]. The corresponding chemical reactions are

\[
\begin{align*}
\text{SF}_6 + e^- & \rightarrow \text{SF}_6^- + F \\
\text{CF}_3\text{Br} + e^- & \rightarrow \text{Br}^- + \text{CF}_3 \\
\text{Ni(CO)}_4 + e^- & \rightarrow \text{Ni(CO)}_3^- + \text{CO}
\end{align*}
\] (2.16)

which follow the dissociation electron attachment reactions of type IV.

At the time of release into the \( F \)-region, these chemical attach electrons. Heavy negative ion clouds are created as a by-product of the electron attachment reaction, and they produce an enhancement in the plasma Pederson conductivity. Eventually, the heavy negative ions and the background \( O^+ \) will neutralize each other and a depletion in the plasma Pedersen conductivity will result [Scales and Bernhardt, 1991]. The mutual neutralization reaction provides a useful ground-based optical diagnostic tool since the neutralization process leaves the oxygen in the excited state.

The electron attachment chemical release has two important advantages over the barium release: no sunlight is required for the chemical reaction to take place and a depletion rather than an enhancement is created. The reduction in plasma densities is often referred to as an \textit{ionospheric hole}. Mendillo [1988] classified three types of ionospheric holes:

- Snow Plow effects
Figure 2.8: Snow plow effect following the passage of large rocket. (After Mendillo [1988].)

Immediately following a neutral gas release, the dense neutral cloud pushes aside the background plasma. An example of this perturbation is given in Figure 2.8. Usually the lifetime of this effect is brief, a few tens of seconds.

- Plasma depletions

Rapid conversion of the dominant ionospheric ion (O⁺) to a molecule ion that quickly recombines with an electron, depicted in Figure 2.9. This is Class III reactions.

- Electron attachment/density depletion

A recently conducted sounding rocket campaigns, NICARE 1 and 2 have been able to create a 30 km diameter electron depletion by releasing attachment materials. Electron density was reduced from $10^5$ cm$^{-3}$ to 15 cm$^{-3}$ in less than 0.1 seconds [Bernhardt et al., 1991 and Ganguli et al., 1993]. This creates a separation between a heavy negative ion plasma from the ambient plasma by a very thin boundary layer of 100 m or less.

Exciting new observations of small scale electron density irregularities were obtained from CRRES [Scales et al., 1994]. These irregularities were observed in the boundary layer in the form of electron density cavities and spikes just within several 0.1 seconds after the release.

In the subsequent chapters, we will lay down the foundations that will be useful in de-
Figure 2.9: Large-scale ionospheric plasma depletion caused by the launch of Skylab in 1973. (After Mendillo [1988].)

Developing a simulation model for ionospheric plasma evolution caused by artificial chemical releases such as barium releases and electron attachment material releases. Numerical simulations of plasma evolution have been extensively studied; however, we will use a different numerical method in our simulation model and use it to simulate some of the ionospheric processes created by artificial chemical releases. We intend to show that, with this new approach, our simulation model and results agree with the observations and also theoretical and numerical studies discussed in this chapter.
3 Basic plasma principles

In the previous chapter we have seen that the ionosphere behaves as as an ionized gas. We shall see in this chapter that the ionosphere possesses certain characteristics that allows it to be classified as a plasma. A plasma is a gas of charged particles which are coupled to each other through their self-consistent electric and magnetic fields. The kinetic energy of a typical particle in a plasma is usually much higher than the potential energy due to its nearest neighbor. So, elements of plasma exert a force on one another even at large distances.

This chapter will review some of the basic principles of plasma theory. We shall discuss some of the physical parameters of plasma such as Debye length, plasma frequency and cyclotron radius. The kinetic and fluid descriptions of plasmas will be presented. Emphasis is placed on a set of equations that describes the fluid behavior of plasmas. Our aim is to define and develop the important concepts that explain plasma evolution described in the subsequent chapters.

3.1 Debye shielding

One of the important behaviors of a plasma is its tendency to maintain charge neutrality. The interaction between charged particles that makes up a plasma is governed by Coulomb forces. An slight imbalance in the charge neutrality gives rise to a much stronger potential energy (due to Coulomb forces) compared to the thermal energy of the particle. Charged particles will move in such a way as to reduce the potential difference and restore the neutrality. If, however, the plasma is subjected to a strong external electric field, the charged particles will adjust themselves so that the majority of the plasma is shielded.

The concept of shielding can be explained as follows [Chen, 1984]. Consider a spherical test charge \(+Q\) in an infinite, uniform plasma. The test charge repels all other ions and attracts all electrons. We assume that it takes a much longer time for the test charge to be neutralized than for each charged particle to reach thermal equilibrium. Thus, around our test charge the electron density \(n_e\) increases and the ion density \(n_i\) decreases. The
test charge gathers a shielding cloud that tends to cancel its own charge. At equilibrium, the distribution of the electrons and ions in the presence of a potential energy $\phi_e$ and $\phi_i$, respectively, follows the Boltzmann distribution

\begin{align*}
    n_e &= n_0 \exp[q_e(\phi_e - \phi_0)/K T_e] \quad (3.1) \\
    n_i &= n_0 \exp[-q_i(\phi_i - \phi_0)/K T_i] \quad (3.2)
\end{align*}

where $\phi_0$ is the local potential and $n_0$ is the density at large distances from the test charge at which the potential vanishes. $K$ is the Boltzmann constant and $T$ is the temperature in K. We assume that the ion-electron mass ratio is infinite such that the ions do not move but form a uniform background of positive charge. Thus, $\phi_i = \phi_0$. Poisson’s equation relating electric potential $\phi$ to the charge densities due to electrons, ions, and the test charge is

\begin{equation}
    \nabla^2 \phi = -\frac{\rho}{\epsilon_0} \quad (3.3)
\end{equation}

or in spherical coordinates,

\begin{equation}
    \frac{1}{r^2} \frac{d}{dr} (r^2 \frac{d\phi}{dr}) = -\frac{1}{\epsilon_0} q_e (n_i - n_e). \quad (3.4)
\end{equation}

Substituting (3.1) and (3.2) into (3.4) yields

\begin{equation}
    \frac{1}{r^2} \frac{d}{dr} (r^2 \frac{d\phi_e}{dr}) = \frac{1}{\epsilon_0} n_0 q_e \exp[q_e(\phi_e - \phi_0)/K T_e - 1]. \quad (3.5)
\end{equation}

We neglect the region where $q_e \phi_e / K T_e$ is large because the potential rapidly falls off so it does not contribute much to the thickness of the cloud. In the region of interest, where $q_e \phi_e \ll K T_e$, we make the approximation $\exp(q_e \phi_e / K T_e) \approx 1 + q_e \phi_e / K T_e$ and,

\begin{equation}
    \frac{1}{r^2} \frac{d}{dr} (r^2 \frac{d\phi_e}{dr}) = \frac{n_0 q_e^2}{\epsilon_0 K T_e} (\phi_e - \phi_0). \quad (3.6)
\end{equation}

Then the solution to (3.6) is

\begin{equation}
    \phi_e = \phi_0 + \frac{Q}{4\pi \epsilon_0 r} e^{-r/\lambda_D} \quad (3.7)
\end{equation}

where $r$ is the radial distance to $+Q$ and $\lambda_D$ is the Debye length given by

\begin{equation}
    \lambda_D = \sqrt{\frac{\epsilon_0 K T_e}{n_0 q_e^2}}. \quad (3.8)
\end{equation}

Debye length is a measure of the shielding distance. Whenever external potentials are introduced into the system or when local concentration of charge arises, they are shielded
out. In plasma this shielding distance must be much smaller than the dimension L of the system, leaving the bulk of the plasma free of large potentials or fields. Beyond $\lambda_D$, $\nabla^2 \phi$ is very small and $n_i$ is equal to $n_e$. The effect of individual particle is cancelled by the collective effect of particles of the opposite sign. However, a small charge imbalance can give rise to potentials of the order $KT/q_e$. Hence, a plasma is said to be quasineutral, that is, neutral enough to have $n_i \approx n_e$, but not so neutral that all of the electromagnetic forces vanish.

At the F region where $T = 1000 \, ^\circ \text{K}$ and $n_0 = 10^{12} \, \text{m}^{-3}$, the corresponding Debye length is about 2.2 mm.

3.2 Plasma parameter

The concept of Debye shielding is only valid if there are enough particles in the charge cloud, statistically. We can compute the number of particles $N_D$ in a Debye sphere [Chen, 1984] using (3.8) and obtain

$$N_D = n_0 \frac{4}{3} \pi \lambda_D^3. \quad (3.9)$$

$N_D$ is referred to as the plasma parameter. For the case where $T = 1000 \, ^\circ \text{K}$ and $n_0 = 10^{12} \, \text{m}^{-3}$, $N_D = 4.3 \times 10^4$.

3.3 Plasma frequency

Consider two species of charged particles in a plasma, electrons and positively charged ions. Suppose that the electrons are displaced from the uniform background of ions. An electric field will be built up causing the electrons to be pulled back to their original positions as to restore the charge neutrality. Due to their inertia, the electrons will overshoot and oscillate around their equilibrium positions. The frequency of this oscillation is called the electron plasma frequency. The mass of the ions is much greater than the mass of the electrons so that the ions do not have time to respond to the oscillation field of electrons and may be considered as fixed.

In deriving the expression for plasma frequency [Chen, 1984], we will make the following
assumptions: the plasma is infinite; there is no thermal motion; there is no magnetic field; the ions are fixed and are uniformly distributed in space. We will also assume that the electrons only move in the x-direction. If perturbation of charge density \( \rho \) in x-direction by a small distance \( \xi \) is \( \delta \rho = n_0 q_e \frac{\partial \xi}{\partial x} \), we can represent displacement of the \( \mathbf{E} \) field using the Maxwell's equation

\[
\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}
\]

as follows,

\[
\frac{\partial \mathbf{E}}{\partial x} = \frac{n_0 q_e}{\epsilon_0} \frac{\partial \xi}{\partial x} \tag{3.11}
\]

where \( n_0 \) is the density of each species of the charged particles. By integrating (3.11) we obtain

\[
E = \frac{n_0 q_e \xi}{\epsilon_0} \tag{3.12}
\]

and we see that the electric field and, hence, the force \( q_e E \) on each electron are proportional to the displacement \( \xi \). This force is also perceived as the restoring force. The equation of motion from Newton's second law is given by

\[
m_e \frac{\partial^2 \xi}{\partial t^2} + \frac{n_0 q_e^2}{\epsilon_0} \xi = 0. \tag{3.13}
\]

Equation (3.13) describes the oscillation of the electron cloud around the fixed ions. This equation is in a standard form of a harmonic oscillator equation with the characteristic frequency

\[
\omega_p = \sqrt{\frac{n_0 q_e^2}{\epsilon_0 m_e}} \tag{3.14}
\]

where \( m_e \) is the mass of electron. This is the electron plasma frequency. A convenient approximate formula for electron plasma frequency is

\[
f_p = 8.978 \sqrt{n_0}, \tag{3.15}
\]

where \( n_0 \) and \( f_p \) are in \( \text{m}^{-3} \) and Hz, respectively. Note that this frequency depends linearly on the plasma density and inversely on the electron mass. The plasma frequency is high since the electron mass is small. For a typical value in F region where \( n_0 = 10^{12} \text{ m}^{-3} \), \( f_p = 8.98 \text{ MHz} \). We require \( \omega_p \tau_e > 1 \) for a plasma to exist, where \( \tau_e \) is the mean free time between collisions of electrons with neutral atoms. For \( f_p = 8.98 \text{ MHz}, \tau \approx 0.1 \mu s \).
The preceding three sections of this chapter provide us with conditions that must be satisfied before an ionized gas can be classified as a plasma:

- It has to be dense enough that the Debye length is much smaller than other physical dimensions of interest, \( \lambda_D \ll L \).

- There are enough charged particles in the Debye sphere with the radius of Debye length to make the Debye shielding statistically correct, \( N_D \gg 1 \).

- The collisional frequency between the charged and neutral particles is small compared to the typical plasma oscillation frequency \( \omega_p \) such that \( \omega_p \tau_c > 1 \), where \( \tau_c \) is the mean free time between collisions.

Since the ionosphere meets these three conditions, it can be treated as a plasma.

3.4 Charged particles gyromotion in uniform \( \mathbf{B} \) fields

The trajectory of a single particle of charge \( q \) in the electric and magnetic fields is described by the Lorentz force equation:

\[
\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}).
\]  

(3.16)

A complete derivation of charged particles gyromotion can be found in Chen [1984]. Here, we will first consider a uniform magnetic field where there is no applied electric field so that \( \mathbf{F} = q\mathbf{v} \times \mathbf{B} \). The Lorentz force on the particle is normal to both \( \mathbf{v} \) and \( \mathbf{B} \). The particle is constrained to move in a curved path, and since the parallel component \( v_\parallel \) along the \( \mathbf{B} \) field is not affected by the magnetic field, the path will be a circular. The Lorentz force provides a centripetal acceleration \( v_\perp^2/r \) where \( r \) is the radius of the circle. Now, using (3.16) and Newton’s second law of motion we obtain

\[
qv_\perp B = \frac{mv_\perp^2}{r}
\]

(3.17)

where \( m \) is mass of the particle. We solve for \( r \) in the previous equation and define the radius of the orbit as the Larmor radius,

\[
r_L = \frac{mv_\perp}{|q|B}
\]

(3.18)
and the cyclotron frequency or gyrofrequency $\Omega_c$ is

$$\Omega_c = \frac{v_\perp}{r_L} = \frac{qB}{m}.$$  \hfill (3.19)

This frequency is also known as the Larmor frequency. By this convention we note that the cyclotron frequency depends on the charge of the particle involved. Thus, opposite charged particles in the same magnetic field gyrate in opposite direction. Plasmas are diamagnetic. This is another illustration of the fact that plasma particles tend to maintain charge neutrality.

The complete trajectory of a charged particle is the gyration of the particle in a circular orbit superimposed on the movement of guiding center along the magnetic field. The result is a helical motion along the uniform magnetic field line.

If both the electric and magnetic field are present, the particle motion will be the sum of the circular Larmor gyration and the drift of the guiding center. The three dimensional orbit in space is therefore a slanted helix with changing pitch. A constant drift velocity exists perpendicular to both $\mathbf{E}$ and $\mathbf{B}$, also known as the $\mathbf{E} \times \mathbf{B}$ drift. We will derive the $\mathbf{E} \times \mathbf{B}$ drift later in section 3.7.3.

Lorentz force equation can be manipulated to show that the total motion of charged particle consists of constant velocity parallel to $\mathbf{B}$, constant drift velocity perpendicular to both $\mathbf{E}$ and $\mathbf{B}$, and gyration about the magnetic field lines.

To illustrate typical magnitude in the upper ionosphere, we consider an electron and an oxygen ($\text{O}^+$) ion. Let $B = 0.3 \text{ G} \ (= 3 \times 10^{-5} \text{ Wb/m}^2)$ and the temperatures of electrons and ions are 1000 °K and 250 °K, respectively. The corresponding cyclotron frequency and Larmor radius are:

$$\Omega_{ce} = -5.3 \times 10^6 \text{ rad/sec or } f_{ce} = 0.84 \text{ MHz}, \ r_{Le} = 2.3 \text{ cm}$$

$$\Omega_{ci} = 180 \text{ rad/sec or } f_{ci} = 28.6 \text{ Hz}, \ r_{Li} = 2.0 \text{ m}.$$  \hfill (3.20)

Finally, if the particles are in inhomogeneous fields where $\mathbf{E}$ and $\mathbf{B}$ fields vary in space or time, the solution for the guiding center drift becomes too complicated to solve exactly. More details can be found in Chen [1984] and Nicholson [1983]. Mirror confinement in plasma arises from the derivation of particle motion in nonuniform fields. It is the basis for one of two explanations on magnetic fusion and is the reason for the existence of the Earth's magnetosphere.
3.5 Kinetic description of plasma

The motion of particles in real plasma is much more complicated than what we encountered in the last section. The $E$ and $B$ fields are not prescribed but have to be determined by the positions and motions of the charges themselves. The problem is then to find a set of particle trajectories and field patterns such that the particles will generate the fields as they move along their orbits and the fields will cause the particles to move in those exact orbits. Not only it is a self-consistent problem, it must be done in time-varying situation.

One of the solution to our self-consistent problem is to use the plasma kinetic theory. This theory takes into account the motions of all particles. This can be done in an exact way using the Klimontovich equation or the Liouville equation. However, we are mostly interested in certain average or approximate characteristics of the motion, not so much in the exact motion of all particles at all time. This allows us to make a few approximations that lead to Vlasov equation. The Vlasov and Boltzmann equations are considered the fundamental equations of plasma physics. As we shall see later, the exact Klimontovich and Liouville equations are useful as starting points for the derivation of approximate equations that describe the average properties of a plasma. We sketch these derivations below.

The exact Klimontovich equation [Nicholson, 1983] is given by:

$$\frac{\partial N_s(x, v, t)}{\partial t} + v \cdot \nabla_x N_s + \frac{q_s}{m_s} (E^m + v \times B^m) \cdot \nabla_v N_s = 0 \quad (3.21)$$

where $N_s$ is the density of species $s$, $x$ is a three-dimensional configuration space and $v$ is the three-dimensional velocity space. Superscript $m$ denotes microscopic electric and magnetic fields self-consistently produced by the point particles themselves, while $\nabla_x$ and $\nabla_v$ denote spatial derivative and velocity derivative, respectively. Together with Maxwell's equations, (3.21) constitutes an exact description of a plasma. That is, this equation tells us if a particle with infinite density is to be found at a given point $(x, v)$ in phase space. We want to know the number of particles likely to be found in a small volume $\Delta x \Delta v$ of phase space centered at $(x, v)$. Hence, we are more interested in

$$f_s(x, v, t) = \langle N_s(x, v, t) \rangle \quad (3.22)$$

which is the ensemble average of $N_s$. Using similar distribution function for $E$ and $B$ and
ensemble averaging (3.21), we obtain
\[
\frac{\partial f_s(x, v, t)}{\partial t} + v \cdot \nabla_x f_s + \frac{q_s}{m_s} (E + v \times B) \cdot \nabla_v f_s = -\frac{q_s}{m_s} \langle \delta E + v \times \delta B \rangle \cdot \nabla_v \delta N_s \tag{3.23}
\]
where \(\delta N_s = N_s - f_s\), \(\delta E = E^m - \langle E^m \rangle\), and \(\delta B = B^m - \langle B^m \rangle\). This equation is called the plasma kinetic equation. The left side of (3.23) contains only smoothly varying terms in \((x, v)\) space representing collective effects. The right side of (3.23) is sensitive to the discrete-particle nature of the plasma which give rise to collisional effects. But as we have seen in Section 3.3, collisional effects in plasma is much much smaller than the collective effects. Thus, the right side of (3.23) can be neglected and we obtain
\[
\frac{\partial f_s(x, v, t)}{\partial t} + v \cdot \nabla_x f_s + \frac{q_s}{m_s} (E + v \times B) \cdot \nabla_v f_s = 0 \tag{3.24}
\]
which is the Vlasov equation named after Soviet mathematician A. Vlasov in 1945.

The Liouville equation describes plasma as the behavior of system instead of the individual particles as treated by the Klimontovich equation. The Liouville formulation treats three positions and three velocities of each particle as independent variables with time as a parameter, so the phase space \((x_1, v_1) = (x_1, y_1, z_1, v_{x1}, v_{y1}, v_{z1})\). If we have a system of \(N\) particles then we are dealing with \(6N+1\) dimensions. The exact Liouville equation [Nicholson, 1983] is given by:
\[
\frac{\partial N_0}{\partial t} + \sum_{i=1}^{N} v_i \cdot \nabla_x N_0 + \sum_{i=1}^{N} \dot{V}_i(t) \cdot \nabla_v N_0 = 0 \tag{3.25}
\]
where \(N_0\) is the density of the system and \(\dot{V}_i(t) = \frac{2m_i}{q_i} \langle E^m(x_i, t) + v_i \times B^m(x_i, t) \rangle\). The subscript \(i\) refers to the \(i^{th}\) particle. Note that the density \(N_0\) in (3.25) is completely different from the density \(N_s\) in (3.21). The density \(N_s\) is the density of particles in six-dimensional phase space, while \(N_0\) is the density of systems in \(6N\)-dimensional phase space.

The derivation of (3.25) leads to the BBGKY hierarchy which is just as hard to solve as the original equation. The simplest approximation in the BBGKY hierarchy is the Boltzmann’s equation,
\[
\frac{\partial f_i}{\partial t} + v_i \cdot \nabla_x f_i + a \cdot \nabla_v f_i = \frac{df_i}{dt} = \frac{df_i}{dt} \tag{3.26}
\]
where the acceleration \(a = \frac{q_i}{m_i} (E_i + v_i \times B_i)\). The Boltzmann’s equation was originally derived for neutral gases where collisions are usually understood to be binary or head-on collisions. There are certain cases in which head-on collisions are important. A good
example is a weakly ionized gas such as the D layer in the ionosphere where collisions between neutral molecules and charged particles happen frequently (see Figure 2.3). In plasmas, however, particle trajectories can be changed without any physical collisions with other particles. In this situation, Fokker-Planck formulation is used to find the collisional term for Coulombic interactions on the right hand side of (3.26) [Nicholson, 1983]. Many plasmas have negligible collisions so that the right side of the Boltzmann's equation can be ignored. The result is the Vlasov equation as found in (3.24). Hence, Vlasov equation is also known as the collisionless Boltzmann equation. Sufficiently hot plasmas where collisions are negligible can be described by the Vlasov equation. This approximate equation that neglects collisional effects is often called the most important equation in plasma physics.

Compared to the Liouville formulation, Vlasov equation reduces the dimension from 6N+1 to just 7 by giving up exact positions and velocities of all particles. A complete derivation of the Klimontovich and Liouville equations and their approximation to Vlasov formulation can be found in many plasma textbooks such as Nicholson [1983].

Vlasov equation (3.24), together with the Maxwell’s equations, make up a complete description of the behavior of a plasma. It is, however, valid only when collisional effects are unimportant. Since the Vlasov formulation is concerned with individual particle motion, it is considered as a kinetic treatment.

Although Vlasov equation is a simplified version of Klimontovich and Liouville formulations, it remains nonlinear. In most cases, further approximations are required to obtain analytical solution. A well-employed technique is done by linearizing the Vlasov equation and applying small amplitude wave. Using this method we can obtain the plasma dispersion relation which describes the frequency and wavelength of plasma waves. Landau damping arises as a mathematical consequence of the solution of the dispersion relation. It is an important characteristic of collisionless plasma. There are two kinds of Landau damping: linear and nonlinear. Both are independent of dissipative collisional mechanism [Chen, 1984; Nicholson, 1984; and Stiz, 1992].
3.6 Fluid description of plasma

Sometimes it is not necessary to consider the fact that each species consists of particles with different velocities as in the kinetic treatment. Many plasma phenomena observed in real experiments can be explained by a rather simple model used in fluid dynamics. The fluid model neglects the identity of individual particles and only the motion of fluid elements as a whole is taken into account. The advantage of this approach is that it leads to equations in three spatial dimensions and time rather than the seven-dimensional phase space used in Vlasov theory. However, fluid theory neglects velocity-dependent effects such as Landau damping and often wave-particle effects.

3.6.1 Derivation of fluid equations

We can derive the fluid description of a plasma by taking velocity moments of the Vlasov equation (3.24) [Nicholson, 1983] where \( f_s(x, v, t) \) is the number density of particles in the phase space \((x, v)\). If we integrate \( f_s(x, v, t) \) over the entire velocity space, we get the conventional particle density defined by

\[
n_s(x, t) = \int dv f_s(x, v, t). \tag{3.27}
\]

Since \( f_s(x, v, t) \) is a probability distribution, we can also define the ensemble average of any quantity \( g \) as

\[
\langle g \rangle = \frac{\int dv g f_s}{\int dv f_s} = \frac{1}{n_s} \int dv g f_s. \tag{3.28}
\]

Note that the fluid velocity \( V_s \) is

\[
V_s(x, t) = \frac{1}{n_s} \int dv v f_s(x, v, t). \tag{3.29}
\]

We integrate (3.24) over all velocity space and use the identities in (3.27) and (3.29). The first term yields \( \frac{\partial n_s(x, t)}{\partial t} \) and the second term becomes \( \nabla_x \cdot (n_s V_s) \). The last terms of the Vlasov equation vanish during velocity integration. The result is the continuity equation

\[
\frac{\partial n_s(x, t)}{\partial t} + \nabla_x \cdot (n_s V_s) = 0 \tag{3.30}
\]

which describes the evolution of the plasma density.
The force equation is derived by multiplying Vlasov equation (3.24) by \( v \) and integrating over all velocity space which yields

\[
\frac{\partial}{\partial t} \int dv \, \nu f_s + \int dv \, \nu \cdot \nabla f_s + \frac{q_s}{m_s} \int dv \, \nu [(E + v \times B) \cdot \nabla f_s] = 0
\]  
(3.31)

The first term in (3.31) becomes \( \frac{\partial}{\partial t} (n_s V_s) \) by (3.29). The second term yields

\[
\nabla_x \cdot \int dv \, \nu \cdot f_s = \nabla_x \cdot (n_s \langle \nu \nu \rangle)
\]  
(3.32)

by (3.28). The third term is evaluated by an integration by parts, resulting in \( -\frac{2}{m_s} n_s E \). The last term, involving the cross product of \( \nu \) and \( B \) gives us \( \frac{2}{m_s} n_s (V_s \times B) \) by using (3.27) and (3.29). Combining all the terms, (3.31) is written as

\[
\frac{\partial}{\partial t} (n_s V_s) + \nabla_x \cdot (n_s \langle \nu \nu \rangle) = \frac{q_s}{m_s} (E + V_s \times B)
\]  
(3.33)

which is the fluid force equation for species \( s \).

Multiplying (3.33) with the mass \( m_s \) yields

\[
m_s n_s \frac{\partial V_s}{\partial t} + m_s \nabla_x \cdot (n_s \langle \nu \nu \rangle) = q_s n_s (E + V_s \times B)
\]  
(3.34)

The continuity equation (3.30) for \( n_s \) involves function \( V_s \) while the force equation (3.33) involves the function \( \langle \nu \nu \rangle \). Hence, every equation for \( n \) factors of \( \nu \) will require a term with \( n + 1 \) factors of \( \nu \). To find a complete description of a plasma, one needs to take an infinite number of moment equations. In practice, however, we use physical argument to evaluate the term with \( n + 1 \) factors of \( \nu \) rather than using the fluid equation for that term.

We consider a distribution that has a net velocity \( V_s \) in a certain direction and has an isotropic velocity distribution with \( V_s \) such that

\[
\langle \nu \nu \rangle = V_s V_s + \frac{P}{m_s n_s}.
\]  
(3.35)

Using the equation of state of a gas as the approximation for the pressure term \( P \), we have

\[
P = \gamma n KT
\]  
(3.36)

where \( \gamma \) is the ratio of specific heats. In the one-dimensional case \( \gamma = 1 \) for isothermal compression and \( \gamma = 3 \) for adiabatic compression. This pressure force arises from the random motion of particles in and out of a fluid element.
Generally, the last term in (3.35) must be given by \( \frac{1}{n_s \rho_s} \mathbf{P} \) where \( \mathbf{P} \) is the stress tensor whose components \( P_{ij} = m_s n_s (v_i v_j) \) specify both the direction of motion and the component of momentum involved. Equation (3.34) then becomes

\[
m_s n_s \frac{\partial \mathbf{V}_s}{\partial t} + (\mathbf{V}_s \cdot \nabla) \mathbf{V}_s + \nabla \cdot \mathbf{P} - q_s n_s (\mathbf{E} + \mathbf{V}_s \times \mathbf{B}) = 0
\]  

(3.37)

This equation is called the momentum equation since it determines the time rate of change of momentum per unit volume. The momentum equation describes the evolution of the velocity in the plasma. The term \( \frac{\partial \mathbf{V}_s}{\partial t} + (\mathbf{V}_s \cdot \nabla) \mathbf{V}_s \) in (3.37) is referred to as the convective derivative. The first term of the convective derivative represents the change of \( \mathbf{V} \) at a fixed point in space, and the second term represents the change of \( \mathbf{V} \) as the observer moves with the fluid into a region of different \( \mathbf{V} \). This allows us to have an equation for fluid elements fixed in space.

3.6.2 Complete set of equations for fluid description in plasma

We can easily incorporate neutral collisions by adding the term \( m_s n_s (\mathbf{V}_x - \mathbf{V}_0) / \tau \) to the right hand side of (3.37), where \( \mathbf{V}_0 \) is the velocity of the neutral fluid and \( \tau \) is the mean free time before collisions.

In the fluid approximation, we consider the plasma to be composed of two or more interpenetrating fluids, one for each species. In the simplest case, where there is only one species of ion, we need an equation of motion for each the positively charged ion fluid and the negatively charged electron fluid. In a partially ionized gas, an equation of motion is needed for the fluid of neutral atoms. The neutral fluid will interact with the ions and electrons only through collisions. In the absence of collisions, the ion and electron fluids will still interact with each other because of the \( \mathbf{E} \) and \( \mathbf{B} \) fields they generate. The complete set of fluid equations for the many-fluid description of plasma is as follows. For each species \( j \), we have

\[
\frac{\partial n_j}{\partial t} + \nabla \cdot (n_j \mathbf{V}_j) = 0
\]  

(3.38)

\[
m_j n_j \frac{\partial \mathbf{V}_j}{\partial t} + (\mathbf{V}_j \cdot \nabla) \mathbf{V}_j + \nabla \cdot \mathbf{P}_j - q_j n_j (\mathbf{E} + \mathbf{V}_j \times \mathbf{B}) = \text{(collision terms)}
\]  

(3.39)

along with the Maxwell’s equations,

\[
\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}
\]  

(3.40)
\[ \nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (3.41) \]
\[ \nabla \cdot \mathbf{D} = \rho \quad (3.42) \]
\[ \nabla \cdot \mathbf{B} = 0 \quad (3.43) \]

which are related to the continuity and momentum equations through the charge and current densities

\[ \rho = \sum_j n_j q_j \quad (3.44) \]
\[ \mathbf{J} = \sum_j n_j q_j \mathbf{V}_j \quad (3.45) \]

\( \rho \) and \( \mathbf{J} \) are related by the current continuity equation

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \quad (3.46) \]

Simultaneous solution of these equations gives a self-consistent set of fields and motions in the fluid approximation.

There is a significant range of frequencies outside \( 0 \leq \omega < \nu \) where the Vlasov equation does not apply (\( \nu \) denotes the collision frequency). However, we can use the fluid equations with an extra term \( \mathbf{K}_j(\mathbf{x}) \) included in the momentum equation to obtain a set of equations that are valid for low frequencies. Here \( \mathbf{K}_j \) is the term representing the change in momentum of species \( j \) at position \( \mathbf{x} \) due to collisions. When written for both electrons and ions, this set is called the two-fluid model. We can further combine the electron equations and the ion equations to obtain a one-fluid model characterized by:

- mass density \( \rho_M(\mathbf{x}) = m_e n_e(\mathbf{x}) + m_i n_i(\mathbf{x}) \approx m_i n_i(\mathbf{x}) \)
- charge density \( \rho_c(\mathbf{x}) = q_e n_e(\mathbf{x}) + q_i n_i(\mathbf{x}) = q_e (n_i - n_e) \)
- current density \( \mathbf{J} = q_i n_i \mathbf{V}_i + q_e n_e \mathbf{V}_e \quad (3.47) \)
- center of mass fluid velocity \( \mathbf{V} = \frac{1}{\rho_M} (m_i n_i \mathbf{V}_i + m_e n_e \mathbf{V}_e) \)
- total pressure \( P = P_e + P_i \)

This approximation describes the magnetohydrodynamics (MHD) approach. A complete formulation can be found in many plasma physics books [Nicholson, 1983 and Chen, 1984].

### 3.6.3 Incompressible fluid

We can make further simplification of the fluid continuity equation (3.30) for special cases in which the fluid involved is not compressible. We rewrite (3.30) using vector identity
as follows
\[ \frac{\partial n_s(x, t)}{\partial t} + V_s \cdot \nabla n_s + n_s(\nabla \cdot V_s) = 0, \] (3.48)
or equivalently,
\[ \frac{dn_s}{dt} + n_s(\nabla \cdot V_s) = 0, \] (3.49)
where the total time derivative \( \frac{d}{dt} \) is \( \frac{\partial}{\partial t} + V_s \cdot \nabla \). Equation (3.49) states that the rate of change in \( n_s \) moving with the flow is determined only by the divergence of the velocity field. In an incompressible fluid, the velocity field is divergence free so that \( \frac{dn_s}{dt} = 0 \), that is, the mass density of the fluid cannot be created nor destroyed as it moves in time. The continuity equation then becomes
\[ \frac{\partial n_s(x, t)}{\partial t} + V_s \cdot \nabla n_s = 0. \] (3.50)
We shall see an example of simulation of incompressible fluid in chapter 5.

3.7 Electrostatic approximation of ionospheric plasma

Now that we have a complete set of equations to describe plasmas, we may make a final approximation for the F region of the ionosphere (up to 300 km) that will be used in our discussion of plasma evolution in the following chapters. As we shall see, we can make some simplifications that will lead to the electrostatic approximation of plasmas. We will begin by pointing out specific behaviors of the ionosphere in this region that support our argument for using the electrostatic approximation.

3.7.1 The concept of \( \beta \)

It is customary to classify plasmas using the ratio of particle pressure and magnetic field pressure, usually denoted by \( \beta \):
\[ \beta = \frac{\sum nKT}{B^2/2\mu_0}. \] (3.51)
High-\( \beta \) plasmas are common in space and MHD energy conversion which denote that the value of \( B \) is small. We use the earth's magnetic field for \( B \) in (3.51) which acts as a dipole and varies inversely with \( r^3 \) in the Coulomb law. In the ionospheric region of our interest, the magnetic field is large enough so that the magnetic field pressure dominates the particle
pressure. Hence, we can employ low-β plasma models where β is between $10^{-3}$ and $10^{-6}$ [Chen, 1984]. In low-β region, particle motion does not affect the magnetic field so we may assume a uniform field $\mathbf{B}$ and we treat the plasma waves in an electrostatic manner.

### 3.7.2 Electrostatic Maxwell’s equations

As a consequence of the low-β plasma behavior, we have $\frac{\partial \mathbf{B}}{\partial t} = 0$ and hence $\mathbf{E} = -\nabla \phi$. Using this result and the Gauss’s law, we can obtain the Poisson’s equation

$$\nabla^2 \phi = -\frac{\rho}{\varepsilon_0}. \quad (3.52)$$

In an ionized medium such as the ionosphere, very small charge differences can create large electric fields. Thus, a plasma must nearly exhibit charge neutrality ($\frac{\partial \rho}{\partial t} \approx 0$ in current continuity equation) which implies that $\nabla \cdot \mathbf{J} \approx 0$ and the number of electrons per unit volume must be equal to the number of positive ions of all types. Furthermore, the conduction current in Maxwell’s equation $\nabla \times \mathbf{B} = \mu_0 (\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t})$ in (3.41) is much larger than the vacuum displacement current $\varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$ for all frequencies of interest. The displacement current is therefore dropped. It is trivial to show that this last simplification is valid by taking the divergence of the curl of $\mathbf{B}$. The result is the charge neutrality requirement.

We can now rewrite Maxwell’s equation for electrostatic treatment

$$\mathbf{E} = -\nabla \phi \quad (3.53)$$
$$\nabla \times \mathbf{H} = \mathbf{J} \quad (3.54)$$
$$\nabla \cdot \mathbf{D} = -\rho \quad (3.55)$$
$$\nabla \cdot \mathbf{B} = 0 \quad (3.56)$$

These equations along with continuity and momentum equations form the basis for the electrostatic description of plasma.

### 3.7.3 Motion parallel and perpendicular to $\mathbf{B}$

A fluid element is composed of many individual particle so that we may expect the fluid to have drifts perpendicular and parallel to $\mathbf{B}$. Drifts parallel to $\mathbf{B}$ result in the fluid being accelerated along $\mathbf{B}$ under the combined electrostatic and pressure gradient forces. Furthermore, parallel drifts applied to electrons accelerate them to high energies very quickly.
Since electrons cannot leave a region without leaving a large ion charge, the electrostatic and pressure gradient forces on the electrons must be closely in balance [Chen, 1984]. This enforces the neutrality condition in that a plasma has an overriding tendency to remain neutral.

We will consider two types of drifts perpendicular to B. These drifts may be determined from the momentum equation (3.37). The convective derivative term in the momentum equation can be neglected if two conditions are satisfied [Bernhardt et al., 1991]. First, $\frac{\partial v}{\partial t} \cdot v$ must be much less than ion-neutral collision frequency $\nu_{in}$. Second, $v$ must be much less than the speed of sound. Since these conditions are satisfied in the ionospheric region of interest, we now have

$$0 = qn(E + v_\perp \times B) - \nabla p$$

Taking the cross product of (3.57) with B results in

$$v_\perp = \frac{E \times B}{B^2} - \frac{\nabla p \times B}{qnB^2} = v_E + v_D$$

where

$$E \times B \text{ drift: } v_E = \frac{E \times B}{B^2}$$

$$\text{Diamagnetic drift: } v_D = \frac{\nabla p \times B}{qnB^2}$$

These drifts exist in magnetized plasmas that have an external electric field or pressure gradient.

We have now derived a complete set of fluid equations in (3.38), (3.52) - (3.56), and (3.59) - (3.60) that outlines the electrostatic plasma behavior. These equations will be used to describe plasma evolution in chapter 5. However, before we solve this set of equations, we need to develop the appropriate numerical methods. We shall describe these methods in the next chapter.
4 Numerical Simulation Methods

In the previous chapter, we developed the concept of kinetic and fluid equations that are used to describe the behavior of plasmas. The long range nature of the electromagnetic interactions between charged particles in plasmas gives rise to collective modes in which large number of particles move in unison. Various collective modes often interact non-linearly so that only the simplest plasma problems can be solved analytically. In most cases, we would have to settle for linearizing the partial differential equations (PDEs) before they can be solved analytically; or, lesser important phenomena may have to be neglected to simplify the problem to manageable proportions. Neither of these options is ideal to study complicated problems such as the ionospheric irregularities that involve a number of non-linear physical processes. By linearizing, we can predict if instabilities will occur, but we cannot describe how instabilities develop non-linearly.

Numerical simulation offers an alternative means that allows us to follow the dynamics of numerous interacting degrees of freedom and non-linearity of the systems. We can pick out specific physical processes of interest to model, thereby isolating their effects. We can also track very detailed behavior of plasma during the simulation, such as the velocity of a species at a particular time instant. Numerical simulation allows us to freely modify all physical parameters. This advantage is useful in investigating what maybe important underlying mechanisms of plasma.

This chapter deals with numerical methods used for 2-D fluid simulations of plasmas. First, we begin by providing an overview of different types of plasma simulations, followed by descriptions of the physical set-up of our simulation. We then devote the majority of work in this chapter to detailed discussion of spectral methods used to solve the fluid plasma equations. These methods are used to solve specific case of active perturbation in the ionosphere in the next chapter.

4.1 Classification of numerical simulation of plasmas

Computer simulation of plasmas is broadly dictated by the type of equations it solves,
kinetic and fluid descriptions. This classification can be divided further into four types of plasma simulation:

- Fluid and MHD codes
- Vlasov and Fokker-Planck codes
- Particle codes
- Hybrid codes

It is obvious that the first type of simulation, fluid and MHD codes, falls under the fluid description of plasma, while the second and third types utilize the kinetic nature of plasma.

4.1.1 Fluid simulation method

Fluid simulation is the oldest method in plasma simulation. This type of simulation regards the many-body system as being composed of so many particles so that the identity of each individual particles is not as important as the collective nature of plasma particles as the discretization parameter of the plasma becomes small. The equations in the fluid formulation treat the plasma as a continuous, incompressible and inviscid fluid. The discretization of the mathematical model reintroduces particles (fluid elements), which now have the attributes of position and velocity. Special care must be taken to ensure numerical stability due to the presence of the convective terms in the fluid equations. Due to the nature of fluid treatment, fluid simulation methods are generally used to describe the plasma behavior in large space and time scales. Ad hoc transport coefficients and other assumptions must often be inserted to relate microscopic processes into fluid models.

In a magnetized plasma, this fluid formulation is replaced by the magnetohydrodynamic (MHD) model. MHD codes can then be used to simulate the plasma.

4.1.2 Particle simulation

Particle simulation attempts to emulate plasma behavior by following the motions of a large number of charged particles interacting with each other and with externally applied fields. It is the most rudimentary method to simulate a plasma, and in most cases, the particles in the simulation model may be identified directly with the particles in the plasma.
Each particle has a set of attributes such as mass, position, charge, and momentum. The evolution of the processes in the plasma is determined by the laws of interactions between the particles using the discretized version of Maxwell's equations, Newton's second law of motion, and the Lorentz' force equation. Since particle models follow the motion of the plasma on the finest space scale and the most rapid time scale, they are limited to looking at the phenomenon in a relatively small sample of plasma and over relatively short periods of time. Space scales are typically on the order of the Debye length $\lambda_D$, while time scales are on the order of ion or electron plasma period, depending on the application.

There are three principle types of particle simulation model: the particle-particle (PP) model, the particle-mesh (PM) model and the particle-particle-particle-mesh (PPPM) model. The PP method uses the action at a distance formulation of the force using Coulomb's law. It is conceptually and computationally the simplest, but it is also very slow. The PM method, also known as the particle-in-cell (PIC) simulation, treats the force as a field quantity by approximating it on a mesh. The PM method has an enormous speed gain over the PP method, obtained at the cost of a loss of resolution in the potential and force fields. Only those field variations which have wavelengths longer than the spacing of the mesh or grid cell can be accurately represented by the mesh values. PPPM method combines the advantages of the PP and PM methods and enables large correlated systems with long-range forces to be simulated. The PP method is employed to compute the total short-range contribution to the force on each particle and the PM method is used to find the total slowly-varying force contribution. The summation of all force contribution gives the total forces on each particle which is used to update the velocities. Three comprehensive sources on plasma simulation using particles are *Birdsall and Langdon* [1991], *Hockney and Eastwood* [1988] and *Tajima* [1989].

4.1.3 Vlasov and Fokker-Planck codes

Vlasov and Fokker-Planck codes solve the plasma kinetic equations (Vlasov or Fokker-Planck equations) numerically. Collisonal term in the Vlasov equation is usually evaluated using Fokker-Planck equation, hence the name. Vlasov simulation treats phase space as a continuum. This approach avoids statistical errors otherwise present in particle simulation. However, as we have seen in section 3.5, there are seven independent variables associated
with the Vlasov equation that need to be solved from seven nonlinear integral-differential equations. Finding numerical solution to these nonlinear equations is not a trivial task. Furthermore, due to the corrective term in the Vlasov equation, the solution is prone to numerical instabilities. Hence, only a limited number of dimensions over a short time scale can be reasonably simulated using Vlasov and Fokker-Planck codes.

4.1.4 Hybrid codes

As the time scale gets smaller, the overall plasma behavior becomes more like particle, and vice versa. There exist intermediate time scales in which a mixing of the particle-like and the fluid-like behaviors often occurs. Such phenomena is best modeled by a combination of fluid and particle simulations, known as the hybrid code. Hybrid codes have recently gained more importance because they allow the modeling of the intermediate time scales in ionospheric plasma simulations, a study that was not possible before. In hybrid simulation, fluid and particle treatments are applied to different components of a given plasma. This may happen because of the wide range of frequency separated by the mass difference between electrons and ions. Take for example the ion species created through electron attachment chemistry by the neutrals [Scales et al., 1994]. Kinetic effects are important only for the ion species. In this case, electrons may be treated as a massless fluid while treating the ions and neutrals as simulation particles.

Of the four types of plasma simulations described above, the fluid simulation method is most suitable for studying the plasma evolution in the ionosphere that will be discussed in the next chapter. This is because we are mostly interested in the macroscopic processes of the plasma over large time scales; the behavior of individual particles is not important in our problem. The next section outlines our basic fluid simulation model.

4.2 Two-dimensional electrostatic fluid simulation

Simulation of a plasma in two or three spatial dimensions is much more realistic than in one dimensional model. We begin our model by defining our simulation box. Two spatial dimensions $x$ and $y$ are assumed. The plasma length in both $x$- and $y$-directions $L_x$ and $L_y$ are equally divided into the number of grid cells $nx$ and $ny$, respectively. The grid size
is made small enough to resolve necessary details. Thus, there are $n_x+1$ grid points on the x-direction and $n_y+1$ grid points in the y-direction. In practice, we usually make the plasma length to be equal in both spatial directions. The same is true for the number of grid cells. Figure 4.1 shows the typical two-dimensional rectangular grid. We require that $n_x$ and $n_y$ be an integer power of 2 since we will be using base-2 Fast Fourier Transform (FFT) techniques that will be discussed later. Furthermore, $L_x$ and $L_y$ should represent plasma length of a periodic system. We generally use index $i$ to denote grid points in the x-direction, and index $j$ for the y-direction. It is obvious that the parameters $dx$ and $dy$ are simply $L_x/n_x$ and $L_y/n_y$, respectively.

Figure 4.2 illustrates a fluid element that is situated in our electrostatic fluid simulation. In reality, the fluid element is in a 3-D space; however, in our 2-D program, we only account for movement affected by velocities $v_x$ and $v_y$ in the x- and y-directions. A magnetic field $B$ uniform over the whole plasma length exists along the z-direction.

We may perceive the ionosphere as an infinite space along the horizontal direction. Hence, for our purposes, we can represent the ionosphere as periodic systems. Such systems
are necessarily charge neutral in that

$$0 = \int_{\text{period}} \mathbf{E} \cdot d\mathbf{S} = \int_{\text{period}} \rho \, dV$$  \hspace{1cm} (4.1)

as discussed in the last chapter.

This relation holds for plasma regions that are more than a few Debye lengths across and away from the sheath. This condition is fully met by our plasma model. We can, furthermore, impose a doubly periodic boundary condition on our simulation space such that the flux that leaves the region from one side will enter from the other side, for both the x- and y-directions.

In the next chapter we will develop a model of the evolution of plasma cloud enhancement or depletion as it moves across the ionosphere. The evolution can be described by the following four equations [Perkins et al., 1973]:

- **Potential equation:** \[ \nabla \cdot n \nabla \phi = 0 \]  \hspace{1cm} (4.2)
- **Continuity equation:** \[ \frac{\partial n}{\partial t} + \mathbf{v} \cdot \nabla n = 0 \]  \hspace{1cm} (4.3)
- **Maxwell’s equation:** \[ \mathbf{E} = -\nabla \phi \]  \hspace{1cm} (4.4)
- **Momentum equation:** \[ \mathbf{v} = \frac{\mathbf{E} \times \mathbf{B}}{B^2} \]  \hspace{1cm} (4.5)
These equations have been derived in chapter 4. Complete discussion on the exact plasma model and equations will be found in the next chapter.

It can be seen that our model equations are nonlinear. They also contain spatial derivatives. These must be calculated numerically. Numerous ways exist including finite difference, finite element, and spectral methods. In the next section, these methods will be compared and the most suitable numerical method for our model equations will be chosen.

4.3 Spectral method in spatial discretization

Method of discretization of differential equation in a mathematical model can be broadly classified under three headings: finite difference approximation, finite element approximation, and method of weighted residual (MWR). Spectral methods are sometimes viewed as an extreme development of the MWR [Canuto et al., 1988]. The key elements of the MWR are the trial functions and the test functions. The trial functions are used as the basis functions for a truncated series expansion of the solution. The test functions are used to ensure that the trial functions satisfy the differential equation as closely as possible by minimizing the error (residual) in the differential equation produced by using the truncated expression instead of the exact solution. It is very important to have the trial functions and the test functions properly formulated.

Spectral methods differ from finite element and finite difference methods by the kind of trial functions. The trial functions for spectral methods are infinitely differentiable global functions. Finite element methods, on the other hand, divide the domain into a number of sub-domains and a trial function is specified in each sub-domain. Likewise, the trial functions for finite difference methods are also local in character.

Spectral methods has been the natural choice for the type of simulations involving nonlinear terms such as turbulence and transition studies, ocean dynamics, and ionospheric instabilities. The main advantages of spectral method are its accuracy and its ability to handle nonlinear terms. Spectral methods have an infinite order of exponential convergence [Boyd, 1989], and have a substantial speed advantage over finite differencing in that they need less grid points to produce a comparably accurate solution. The superiority of spectral
Figure 4.3: Accuracy comparison of passive scalar convection of a cone: (a) 2nd order Arakawa method, $32 \times 32$ grid; (b) 4th order Arakawa method, $32 \times 32$ grid; (c) same, $64 \times 64$ grid; (d) spectral method, $32 \times 32$ after 1 revolution. (After Orszag [1971a].)

methods over other discretization methods is evident in the widely cited work of Orszag [1971a]. Figure 4.3 shows comparison of numerical tests of scalar convection by a circular flow of a cone. The finite difference (Arakawa) scheme in various orders produces a wake trailing the peak of the cone. Also, the originally sharp cone has been smeared. On the other hand, the spectral method on a comparable sized grid produces a conic shape hardly modified from the original, as it should be.

There are some drawbacks to spectral methods. Computation done using a spectral method is more costly per degree of freedom than that of finite difference methods. However, with the development of fast transform, the cost of using spectral method is comparable to that of finite element schemes while providing higher accuracy. Another restriction to spectral methods deals with the types of suitable geometry. Unless the geometry of the problem is fairly smooth and regular, spectral methods are penalized by heavier losses of
accuracy and efficiency compared to other methods.

Since our simulation model has a fairly smooth, periodic function with periodic boundary conditions, we shall use spectral methods to solve (4.2) - (4.5).

4.3.1 General spectral methods

There are three most commonly used spectral schemes: Galerkin, collocation, and tau versions. Galerkin approach is the most elegant of the MWR since it uses the same test functions and trial functions. They must be infinitely smooth functions which individually satisfies the periodic boundary conditions. The differential equation is enforced by requiring the integral of each test functions multiplied by the residual to be zero. The tau approach are similar to Galerkin methods in the way that the differential equation is enforced. It is essentially a modification of Galerkin method that is applicable to problems with non-periodic boundary conditions. The collocation approach is the simplest method in MWR. The test functions are translated Dirac delta functions centered at special points called the collocation points. The differential equation then has to be satisfied exactly at those points. The collocation points for both the differential equation and the boundary conditions are usually the same as the physical grid points. The earliest application of the spectral collocation method to partial differential equations was made for spatially periodic problems by Kreiss and Olinger in 1972 who called it the Fourier method and by Orszag in 1972 who termed it pseudospectral [Canuto et al., 1989].

We should briefly note here that the terminology of spectral schemes is not too well standardized. In our work, we shall refer to the Galerkin approximation as the spectral Galerkin while collocation approximation will be known as pseudospectral.

Boundary conditions have a crucial role in the application of spectral methods. Incorrect boundary condition treatment in spectral methods may give rise to strong instabilities, whereas in finite difference methods instabilities due to boundaries appear as relatively weak oscillations. On the other hand, spectral methods do not require numerical boundary conditions as the finite difference algorithms do, in addition to the physical boundary conditions required by the PDE [Orszag and Gottlieb, 1980].

In a very simple PDE, such as constant-coefficient hyperbolic equation, the only difference between Galerkin, tau, and collocation approximations is their treatment of the
boundary terms. However, as we shall see, there are significant differences between these approximations in more complicated problems. Our simulation model allows us to use periodic boundary conditions. Hence, from now on we will only be concerned with the spectral Galerkin and pseudospectral approximations.

4.3.2 Spectral methods using Fourier series

Spectral methods, in general, are based on representing the solution to a PDE as a truncated series of a smooth function of the dependent variable [Gottlieb and Orszag, 1971]. Consider a spatially dependent function $u$. This function can be expanded in terms of an infinite sequence of orthogonal functions $\phi_k$ so that $u = \sum_{k=-\infty}^{\infty} u_k \phi_k$. The most familiar approximate solution in solving problems with periodic functions is the Fourier series. In one dimension, this corresponds to the set of functions $\phi_k = e^{ikx}$ which is an orthogonal system over the interval $(0, 2\pi)$. We can introduce the 1-D Fourier coefficients of $u$:

$$\tilde{u}_k = \frac{1}{2\pi} \int_0^{2\pi} u(x)e^{-ikx} dx \quad k = 0, \pm 1, \pm 2, \ldots$$

(4.6)

where $\tilde{u}_k$ is the continuous Fourier transform of $u$. In computer applications, we have to use the discretized version of the Fourier transform. This is obtained by taking $N$ consecutive sampled values of $u(x)$ at

$$x_j = \frac{2\pi j}{N} \quad j = 0, \ldots, N - 1$$

(4.7)

assuming uniform sampling. The set of points $x_j$ is referred to as nodes or grid points. Here $2\pi/N$ is the sampling interval. The discrete Fourier coefficients of a complex-valued function $u$ with respect to these points are

$$u_k = \frac{1}{N} \sum_{j=0}^{N-1} U_j e^{-ikx_j} \quad k = \frac{-N}{2}, \ldots, \frac{N}{2} - 1$$

(4.8)

where $U_j = u(x_j)$. The inverse Fourier coefficients are

$$U_j = \sum_{k=-N/2}^{N/2-1} u_k e^{ikx_j} \quad j = 0, \ldots, N - 1.$$  

(4.9)

The Discrete Fourier Transform (DFT) maps $N$ complex numbers $U_j$ into $N$ complex numbers $u_k$. It is easy to see that as the number of input data increases, the output frequency
spectrum becomes closer to the continuous spectrum. Hence, we may be interested to know how much computation is involved in computing an N-point DFT. For many years, people thought that DFT was an \( O(N^2) \) process. However, Danielson and Lanczos showed in 1942 that a DFT of length \( N \) can be expressed as the sum of two discrete Fourier transforms, each of length \( N/2 \). Thus, the older DFT method of simply brute multiplication executed some redundant multiplications. In fact, DFT can be computed in \( O(N \log_2 N) \) operations with an algorithm called the Fast Fourier Transform (FFT) [Press \textit{et al.}, 1993]. The difference between \( N \log_2 N \) and \( N^2 \) is huge, especially if \( N \) is large or if the transform is taken over more than one dimension. In most cases, FFT requires \( N \) to be an integer power of 2 for an efficient binary manipulation. The derivation of FFT is quite complicated. Thus, we leave it to many good digital signal processing texts.

Similar to the 1-D case in (4.8) and (4.9), we define the 2-D DFT coefficients

\[
u_{k_1,k_2} = \frac{1}{N_1} \frac{1}{N_2} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} U_{j_1,j_2} \exp(-ik_1 x_{j_1}) \exp(-ik_2 x_{j_2})
\]

where \( k_1 = \frac{-N_1}{2}, \cdots, \frac{N_1}{2} - 1 \) and \( k_2 = \frac{-N_2}{2}, \cdots, \frac{N_2}{2} - 1 \) are the corresponding wavenumbers. The inverse Fourier coefficients are

\[
U_{j_1,j_2} = \sum_{k_1=-N_1/2}^{N_1/2-1} \sum_{k_2=-N_2/2}^{N_2/2-1} u_{k_1,k_2} \exp(ik_1 x_{j_1}) \exp(ik_2 x_{j_2})
\]

defined at discrete grid points \( x_{j_1,j_2} \) where \( j_1 = 0, \cdots, N_1 - 1 \) and \( j_2 = 0, \cdots, N_2 - 1 \).

We can now apply the general Fourier spectral method to the set of fluid equations we need to solve in 2-D. Note that using spectral method in space reduces the a PDE into an ordinary differential equation (ODE). Careful treatment needs to be administered for the nonlinear terms in the fluid equations. As is well known, when spectral methods are used, products in real space give rise to convolution sums in Fourier space; derivation in real space domain is equivalent to multiplication by the algebraic factor \( ik \) in the Fourier domain. Furthermore, Laplacian (\( \nabla^2 \)) corresponds to \(-k^2\) in transform space. These operations will be considered carefully. We will also see how they are treated differently in the spectral Galerkin and pseudospectral approximations.

### 4.3.3 Differentiation

Differentiation using spectral methods, such as computing the gradient of density \( n \)
depends upon in which space it is performed. Taking derivative in the Fourier transform space consists of simply multiplying each Fourier coefficients with the imaginary unit times the corresponding wavenumber. This could be expressed analytically by

\[
\mathcal{F}\left\{\frac{du}{dx}\right\} = i k u_k
\]

(4.12)
in two-dimensional space or higher, where \( \mathcal{F} \) represents the Fourier Transform operation and \( k \) is the Fourier wavenumber. This is the true spectral derivative which we refer to as the Fourier Galerkin derivative.

Differentiation in physical space is based upon the values of the function to be differentiated at the collocation points. Discrete Fourier coefficients are found for these values and are multiplied by \( i k \). The results are consequently transformed back into the physical space. The values of the approximate derivative at the grid points make up the Fourier collocation (pseudospectral) derivative. In general, collocation derivative is not equal to the Fourier Galerkin derivative. However, Canuto et al. [1989] proved that collocation differentiation is spectrally accurate.

Computationally, the evaluation of collocation derivative and Galerkin derivative is comparably equal. The operation count is \( O(N \log_2 N) \) in 1-D. However, since we plan to have all the calculations done in Fourier space, it is just as easy to perform all differentiations using the Fourier Galerkin method. Furthermore, phase errors are avoided if spectral Galerkin method is used to compute derivatives [Orszag, 1971b].

The Laplacian \( (\nabla^2) \) follows directly from derivative operation since \( \nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \) in 2-D. The Laplacian operator in the Fourier space becomes a multiplication of each Fourier coefficients with \( -k^2 \), where \( k^2 = k_x^2 + k_y^2 \) in 2-D.

In performing derivatives and Laplacians on a computation discrete grid, \( k \) and \( k^2 \) are actually replaced by \( \kappa \) and \( K^2 \) [Birdsall and Langdon, 1991], respectively, where

\[
\kappa = k_1 \left[ \frac{\sin(k_1 \Delta x_1)}{k_1 \Delta x_1} \right] k_1 + k_2 \left[ \frac{\sin(k_2 \Delta x_2)}{k_2 \Delta x_2} \right] k_2
\]

(4.13)

and

\[
K^2 = k_1^2 \left[ \frac{\sin \frac{k_1 \Delta x_1}{2}}{\frac{k_1 \Delta x_1}{2}} \right]^2 + k_2^2 \left[ \frac{\sin \frac{k_2 \Delta x_2}{2}}{\frac{k_2 \Delta x_2}{2}} \right]^2
\]

(4.14)
in two-dimensions. This discreteness effect can be explained by considering the derivative operator and the Laplacian operator (in 1-D for simplicity) as

\begin{align}
\frac{du}{dx} & \rightarrow \frac{u^{n+1} - u^{n-1}}{2\Delta x} \\
\frac{d^2u}{dx^2} & \rightarrow \frac{u^{n+1} - 2u^n + u^{n-1}}{(\Delta x)^2}
\end{align}

(4.15) (4.16)

obtained by finite differencing. Then substituting (4.8) into the Fourier transform of (4.15) and (4.16) and specializing them for the 2-D case, we obtain the results in (4.13) and (4.14). \(\kappa\) and \(K^2\) approach the continuous result \(k\) and \(k^2\), respectively, as the grid becomes finer, \(k\Delta x \to 0\).

Orszag [1971a] compared the Galerkin derivative results with the derivatives obtained by finite difference methods on similar grids. He found that the finite difference schemes have lagging phase errors, while the Fourier expansion scheme has only leading phase errors for finite \(\Delta t\) that disappears in the limit \(\Delta t \to 0\).

4.3.4 Convolution

Nonlinear terms such as the ones found in (4.2) and (4.3) need a careful treatment by the spectral method. This is due to the property of the Fourier transform which states products in real space give rise to convolution in Fourier space. Thus, we need a specific method that can evaluate Fourier-space convolution effectively and accurately.

Consider the evaluation of the one-dimensional convolutional sum

\[ w_k = \sum_{p+q=k} u_p v_q \quad |p|, |q| \leq N/2 \]  

(4.17)

where \(u_p\) and \(v_q\) are generally complex, and \(p, q, k\) denote the Fourier mode numbers. Direct summation of (4.17) takes \(O(N^2)\) operations in 1-D, compared to only \(O(N)\) operations if we were to use a finite difference algorithm for the nonlinear terms. If we use FFT, the number of operations is improved somewhat to \(O(N \log_2 N)\) as we have seen earlier. However, the number of operations still gets very large very fast as we go to higher dimensions and larger \(N\). The convolution using Fourier Galerkin is not very efficient. The feature that makes direct evaluation of (4.17) inefficient is their nonlocality: \(w_k\) depends on \(u_p\) and \(v_p\) for \(\max(k - N/2, -N/2) < p < \min(k + N/2, N/2)\).
This nonlocality can be avoided by using suitable discrete Fourier transforms to express \( w_k \) as local product of Fourier-transformed fields. Orszag [1971b] was the first to develop the pseudospectral method. The approach is to use inverse DFT given in (4.9) to transform \( u_p \) and \( v_q \) to \( U_j \) and \( V_j \) in physical space. \( U_j \) and \( V_j \) are defined at exactly \( N \) points of \( x_j = 2\pi/N \) for \( j=0,\ldots,N-1 \). We can define \( W_j \) as the local product \( U_j V_j \) of the transform fields, and then use the DFT to return to the Fourier space. The overall convolution sequence can be summarized by

\[
\begin{align*}
  u_k, v_k \xleftarrow{\text{FFT}} & U_j, V_j \xrightarrow{\chi} W_j \xrightarrow{\text{FFT}} \hat{w}_k
\end{align*}
\]

We find that

\[
\hat{w}_k = \frac{1}{N} \sum_{j=0}^{N-1} U_j V_j e^{-ikx_j}
\]

\[
= \frac{1}{N} \sum_{j=0}^{N-1} \left( \sum_{|p| \leq N/2} u_p e^{-ipx_j} \sum_{|q| \leq N/2} v_q e^{-iqx_j} \right)
\]

\[
= \sum_{|p|,|q| \leq N/2} u_p v_q \frac{1}{N} \sum_{j=0}^{N-1} e^{-i(p+q-k)x_j}.
\]  

(4.18)

where \( \hat{w}_k \) is the aliased convolution sum. Using the discrete transform orthogonality relation

\[
\sum_{j=0}^{N-1} e^{-i(k-p)x_j} = \begin{cases} 
N & \text{if } k \equiv p \pmod{N} \\
0 & \text{otherwise},
\end{cases}
\]  

(4.19)

Equation (4.18) can be rewritten as

\[
\hat{w}_k = \sum_{p+q=k} u_p v_q + \sum_{p+q=k+N} u_p v_q
\]

\[
= w_k + \sum_{p+q=k+N} u_p v_q
\]  

(4.20)

where \( w_k \) is the true convolution term given in (4.17). The second term on the right-hand side of (4.20) originates from the property that \( \exp[i(k \pm N)x_j] = \exp[ikx_j] \) for all integral \( j, k \) such that the discrete grid points \( x_j \) cannot distinguish the wave vector \( k \) and its aliases \( k \pm N, k \pm 2N, \text{etc.} \). For any \( k < N/2 \), there always exists a nonzero aliasing error. Equation (4.20) is applicable only for a 1-D case. Derivation for higher dimension cases, although is much more involved, is similar to that of the 1-D case. In three dimensions,
in addition to the singly-aliased term on the right-hand side of (4.20), there are two other
singly-aliased contributions, three doubly-aliased terms and one triply-aliased contribution.
Aliasing errors usually, but not always, lead to numerical instabilities; they, however, always
lead to inaccuracies, especially for high \(k\) modes.

In order to be as accurate as the true spectral Galerkin convolution, we need to eliminate
the aliasing terms. There are several standard techniques used to remove aliasing and hence,
we could be closer to computing (4.20) exactly.

**Aliasing removal by zero-padding**

The key to this dealiasing technique is to append zeroes to \(u_k\) and \(v_k\) that are already
defined for \(|k| \geq N/2\). We extend this spectral fields by setting \(u_k = v_k = 0\) for \(|k| \geq N/2\).
Then we find \(2N\)-point transform \(\tilde{U}_j\) and \(\tilde{V}_j\) of the extended spectral fields. Finally, we
compute the inverse \(2N\)-point transform of the local product \(\tilde{U}_j \tilde{V}_j\). The result should be
aliased-free \(\tilde{w}_k\) or simply \(w_k\) as obtained using spectral Galerkin.

The disadvantage of using this algorithm is that the transforms must be performed on
\(2N\) rather than \(N\) points. Hence, in \(p\) space dimensions, array \(2^p\) larger than either \(u_k\) or
\(v_k\) must be stored [Orszag, 1971b]. This method is costly for large \(N\) or high \(p\) dimensions.

**Aliasing removal by truncation**

The previous zero-padding can be improved by using an \(M\)-point discrete transform
where \(M \geq (3/2)N\). This technique is sometimes referred to as the 3/2-rule and it requires
an FFT that can handle prime factors of 3. If only integer power of 2 FFT is available, this
dealiasing technique can be implemented by choosing \(3 \frac{1}{2} N\) to be a power of 2.

The disadvantage of this method is that in order to have alias-free convolution, only \(2/3\)
of all the \(k\) modes are used in computation. That means 33% of information carried by the
\(k\) modes is thrown away.

**Aliasing removal by grid or phase shifting**

Another method to remove the aliasing terms employs phase shifts. We first define the
phase shifted discrete transform in the same form of (4.9) such that

\[
\tilde{U}_j = \sum_{k=-N/2}^{N/2-1} u_k e^{i k (x_j + \Delta)} \tag{4.21}
\]
\[ \hat{V}_j = \sum_{k=-N/2}^{N/2-1} v_k e^{ik(x_j + \Delta)} \]

which are just the transforms on a grid shifted by the factor \( \Delta \) in physical space. We then compute the local product of \( \hat{U}_j \hat{V}_j \) and inverse transform it back to the Fourier space. The result of the convolution of the shifted grid is

\[ \bar{w}_k = \frac{1}{N} \sum_{j=0}^{N-1} \bar{U}_j \bar{V}_j e^{ik(x_j + \Delta)} \]

while the result for the unshifted grid is given in (4.18). Note that for the 3-D case, both the shifted and unshifted convolution sum contain contributions from singly-, doubly-, and triply-aliased terms. With some manipulations [Orszag, 1971a], and letting \( \Delta = \pi/N \), \( i.e., \) shift of half a grid cell, we find that for a 3-D model

\[ w(k) = \frac{1}{2}[\hat{w}(k) + \hat{w}(k)] - \sum_n w(k + Nn) \quad ||k|| < N/2 \]

where the 1-D case of \( w(k) \) given by (4.17) is the nonaliased convolution and \( n=(n_1, n_2, n_3) \) have one zero component and two components independently \( \pm 1 \). The first term on the right-hand side of (4.24) represents the transform of the shifted and unshifted local products. The summation term on the right-hand side of (4.24) contains the aliasing we wish to remove.

The second term on the right-hand side of (4.24) involves only doubly-aliased interactions. Therefore, using two types of grids shifted by a half grid cell eliminates all the singly- and triply-aliased terms. Patterson and Orszag [1971] and Orszag [1971a] showed that if the set of allowable modes used in the convolution sum is reduced, there will not be any modes that contribute to the doubly-aliased terms. Patterson and Orszag [1971] presented an algorithm to truncate Fourier modes that lie outside a sphere of radius \( \alpha K \) where \( K = N/2 \) and \( \alpha \geq \frac{2\sqrt{2}}{3} \approx 0.94 \). Orszag [1971a] made an improvement to the previous algorithm by truncating the spectral representation not to the spectral region with radius \( \alpha K \), but to a region bounded by an 18-sided polyhedron. This region contains roughly 27% more modes than that of the spherical region. These two algorithms are equally efficient. Some people opt for the polyhedron region because it carries more modes. However, we choose to use the modes within the sphere radius \( .94 K \) since it is much simpler to implement and since lacking the extra 27% modes did not affect the result of our simulation.
Truncation of the modes outside of .94K is accomplished by using a filter of the form 
[Birdsall and Langdon, 1991]

\[ SM(k) = \exp(-\alpha^n) \]  \hspace{1cm} (4.25)

where \( \alpha = k/k_{max} \). The one-dimensional form of \( SM(k) \) is plotted in Figure 4.4. This filter attenuates the short wavelengths (high \( k \) modes). By specifying \( k_{max} \) and \( n \), we choose the shape of filter that can best truncates the unwanted Fourier modes.

4.3.5 Comparison of spectral and pseudospectral approximation

Pseudospectral approximations have several advantages over spectral Galerkin schemes, including:

- evaluation of convolution using spectral method requires at least twice the number of FFTs as the corresponding pseudospectral method.

- pseudospectral codes are considerably simpler than spectral codes.

- pseudospectral approximation applies to a broader range of problems than spectral Galerkin approximation.

Pseudospectral method sometimes cannot eliminate aliasing completely. Thus, the main drawback of pseudospectral approximation is that it may be more susceptible to numerical instability and inaccuracy due to the aliasing terms. In some simple model problems, Orszag [1972] found that both approximations give similar errors, despite the inclusion of aliasing terms in pseudospectral approximation.

It is necessary that our simulation code has the most efficient spatial solver possible because the spatial solver maybe called a number of times during each time step. Therefore, using the Fourier Galerkin method in spite of its high accuracy is not cost/time effective for our simulation.

4.4 Time integration

In most application of spectral methods to PDEs, the spectral or pseudospectral method is used for spatial discretization, but conventional finite differences are used in temporal
Figure 4.4: Attenuation filter profiles for various values of $k_{max}$ and $n$. (a) $k_{max} = 64$; (b) $k_{max} = 56$; (c) $k_{max} = 48$; (d) $k_{max} = 40$. x-axis represents the Fourier wavenumber. $n = 3$ is the top curve, $n = 4$ is the second curve from the top, and so on.
discretization. We may use finite difference methods for time integration, because we do not have to deal with nonlinearities that may cause inaccuracies and other problems (as in the spatial integration). Furthermore, finite differencing is simple and easy to implement.

The finite difference method is derived from the Taylor series expansion of a function about a particular point

\[ n(t_0 + \Delta t) = n(t_0) + \Delta t \left( \frac{dn}{dt} \right)_{t_0} + \frac{(\Delta t)^2}{2!} \left( \frac{d^2n}{dt^2} \right)_{t_0} + \cdots \]  \hspace{1cm} (4.26)

where \( \Delta t \) is the finite time step. The first derivative is approximated by

\[ \left( \frac{dn}{dt} \right)_{t_0} \approx \frac{n(t_0 + \Delta t) - n(t_0)}{\Delta t} + O(\Delta t). \]  \hspace{1cm} (4.27)

This is known as the forward-difference method. Similarly, we can replace \( \Delta t \) by \(-\Delta t\) to obtain the backward-difference equation.

\[ \left( \frac{dn}{dt} \right)_{t_0} \approx \frac{n(t_0) - n(t_0 + \Delta t)}{\Delta t} + O(\Delta t). \]  \hspace{1cm} (4.28)

The forward and backward finite difference methods by themselves are unstable [Tajima, 1989]. However, as we will discuss, some manipulations can be done to improve the stability.

We can write, for example, (4.3) in a typical evolution equation form

\[ \frac{dn}{dt} = f(n, t) \]  \hspace{1cm} (4.29)

where \( f(n, t) \) is calculated using the spectral and pseudospectral methods. \( n \) is actually the density expressed in Fourier space. We can discretize this equation by putting it in the finite difference form.

In this section, we will discussed some of the standard methods used to solve the temporal dependance of ODEs. We will then choose the time discretization method most appropriate for our application.

*Euler’s scheme*

Euler’s method is the simplest finite difference procedure for dealing with the first-order differentiation. The formula for the Euler’s method is

\[ n^{t+\Delta t} = n^t + \Delta t f(n^t) \]  \hspace{1cm} (4.30)
which advances the density from time $t$ to time $t + \Delta t$, but only uses derivative information taken at time $t$. This means that the step’s error is only one power of $\Delta t$ smaller than the correction, so that $O(\Delta t)$ needs to be added to (4.30).

Euler’s method is not recommended for practical use for several reasons. This method is neither very stable nor very accurate when compared to some other methods run at the equivalent step size. Furthermore, in solving the continuity equation, if for some reason the density becomes negative, the solution will behave badly and may lead to numerical instabilities.

**Leapfrog scheme**

Leapfrog method is derived by subtracting the Taylor representations of the forward and backward expansions about $t_0$. This second-order, two-step scheme is given by

$$n^{t+\Delta t} = n^{t-\Delta t} + 2\Delta t f(n^t)$$ (4.31)

This method is sometimes called the center-difference method. It is more accurate than the Euler’s scheme because it uses derivative information at three points simultaneously. Leapfrog integration method is graphically shown on Figure 4.5. Since it is second-order accurate, the error is proportional to $(\Delta t)^2$. Leapfrog method is also explicit and easy to implement.

However, there are some drawbacks to the leapfrog method. One difficulty with the leapfrog method is that the solution is subject to temporal oscillation with period $2\Delta$. This arises from the extraneous solution to the temporal difference equations. Even-odd instabilities or weak instabilities may arise when the odd and the even time steps used in
the leapfrog scheme become uncoupled. Furthermore, when pseudospectral method is used in space and leapfrog is used in time, errors in leapfrog reside entirely in the phases of individual components. The amount of phase error increases with the Fourier wavenumber of the component and with the length of the time interval [Canuto et al., 1989].

There are improvements to the leapfrog scheme. For example, implicit leapfrog method in the following form is absolutely stable.

\[ n^{t+\Delta t} = n^{t-\Delta t} + 2\Delta t f\left(\frac{n^{t+\Delta t} + n^{t-\Delta t}}{2}\right) \]  \hspace{1cm} (4.32)

However, it is very hard to implement because it involves matrix decomposition.

The temporal oscillation can be controlled by every so often averaging the solution at two consecutive time levels. Even-odd instabilities can be abated by implementing a fictitious diffusion term.

The leapfrog-trapezoidal method [Zalesak, 1979] improves the performance of leapfrog method by strongly damping the computational mode generated in the leapfrog step, thereby also removing the even-odd instabilities. The leapfrog-trapezoidal scheme is as follows:

\begin{align*}
\text{step 1:} & \quad n^t = n^{t-\Delta t} + 2\Delta t f(n^t) \hspace{1cm} (4.33) \\
\text{step 2:} & \quad f(n^*) = \frac{1}{2}(f(n^t) + f(n^t)) \hspace{1cm} (4.34) \\
\text{step 3:} & \quad n^{t+\Delta t} = n^t + \Delta t f(n^*) \hspace{1cm} (4.35)
\end{align*}

where \( n^t \) is the estimate of \( n^{t+\Delta t} \). The leapfrog-trapezoidal integration sequence is graphically shown in Figure 4.6. Time centering in (4.33) and (4.35) guarantees second-order accuracy in time.


**Runge-Kutta methods**

Many Runge-Kutta time discretizations have been applied in combination with spectral methods for PDEs. Runge-Kutta methods move a solution over an interval by combining the information from several Euler-style steps (each involving one evaluation of the right-hand side of (4.29)). Then the information obtained is used to match a Taylor series expansion. Runge-Kutta method is stable and is more accurate than the leapfrog; it is fourth-order accurate. However, it does require extra storage and extra time to evaluate additional information at the intermediate steps.

**Predictor-corrector methods**

Predictor-corrector methods store the solution along the way, and use those results to extrapolate the solution one step advanced. They then correct the extrapolation using derivative information at the new point. The most common predictor-corrector methods are the Adams-Bashford-Moulton schemes. The Adams-Bashfort part is the predictor while the Adams-Moulton part is the corrector. Predictor-corrector methods have good stability properties and are best for smooth functions.

**Richardson extrapolation methods**

Richardson extrapolation uses the idea of extrapolating a computed result to the value that would have been obtained had the step size been much smaller than it actually was. Combined with the Bulirsch-Stoer method, Richardson extrapolation schemes can obtain high-accuracy solutions to ODEs while minimizing the computational effort.

The list of methods given here is simply samples of discretization methods we considered in our simulation work. It is by no means an exhaustive list. Our objective is to have the simulation run efficiently and produce as accurate results as we can achieve. Accuracy and stability are especially important in our case because we want to show the nonlinear effects throughout our simulation run, but we do not want the nonlinearities to create numerical instabilities.

After many simulation runs, we decided that a combination of leapfrog and leapfrog-trapezoidal used in time integration works best. The trapezoidal step is implemented every eighth leapfrog iteration. The simple Euler's method would take the shortest time to run,
but it would also become numerically unstable very quickly. On the other hand, Runge-Kutta and Richardson extrapolation methods would provide a better accuracy. However, it would also take much longer to complete one simulation run.

4.5 Numerical accuracy requirements

There are a number of requirements we should heed when we set up a simulation model for a plasma. The first one is the requirement of time step size as dictated by the famous Courant-Friedrichs-Lewy stability criterion [Tajima, 1989], often simply called as the Courant condition. Applied to the simulation grid size and time step size, we need to have

\[
\frac{|v| \Delta t}{\Delta x} \leq 1
\]  

(4.36)

Furthermore, our numerical model has to fulfill the plasma criterions. That is, the Debye length must be small compared to other physical dimensions of interest. In simulations, Debye length is at least of the order of a grid cell. However, the grid size has to be small enough to avoid aliasing. From sampling thory,

\[
k_{\text{max}} \Delta x \leq \pi
\]  

(4.37)

where \( k_{\text{max}} \) is the maximum wavenumber in the simulation. Plasma has to be long enough to allow events with smallest wavenumber to occur,

\[
k_{\text{min}} \geq \frac{2\pi}{L}
\]  

(4.38)

where \( L \) is the length of the system, or \( Lx \) and \( Ly \) in our case.

In the chapter 5 we will use the numerical methods developed in this chapter for our simulation model of the plasma evolution. The Fourier spectral/pseudospectral methods will be implemented to solve a set of electrostatic fluid equations used in studying important ionospheric processes. We shall discuss the results and plots generated by our simulation model in the next chapter.
5 Numerical simulation of some cases of ionospheric irregularities

In the previous chapters we have developed a set of equations that describes plasma fluid behavior that may be applied to the study of macroscopic ionospheric dynamics. We are now in the position to implement the numerical methods developed in the last chapter to study important ionospheric processes. Our goal is to simulate artificial ionospheric irregularities produced by chemical releases by using the spectral and pseudospectral methods. The novelty of our simulation work lies in the fact that we use spectral/pseudospectral methods in our spatial discretization to model plasma density enhancements and depletions caused by chemical releases. This work will lay the groundwork for more realistic 3-D simulations using spectral methods in future investigations.

This chapter is organized as follows. First, we use the fluid equations that were derived in chapter 3 to develop a theoretical model to describe the evolution of artificial irregularities. Second, the 2-D simulation model used to solve these theoretical equations is presented. We verify the accuracy of our model by providing test results of the numerical algorithms. Finally, we present the simulation runs and discuss our results. Comparisons with other work are also presented.

5.1 Theoretical model

In modeling the irregularities caused by artificial chemical releases, we consider the late time behavior, after all the released materials have been used up and depletion or enhancement (depending on the type of released chemical) has formed. The dynamical properties of these irregularities mimic those of the ionospheric F region irregularities such as the equatorial spread F. These properties can be described accurately using a fluid model comprised of neutral, electron, and ion components. In this section we will we use the fluid equations that were derived in chapter 3 to develop a model in real space to describe the evolution of artificial irregularities. We will briefly discuss the $\mathbf{E} \times \mathbf{B}$ instability which is an important instability associated with the structuring of ionospheric plasma. Next, we
will transform this model into its Fourier space representation. We then discuss the idea of energy conservation in the simulation model.

5.1.1 Theoretical model in real space

The density of the neutral fluid is many orders of magnitude larger than the electron or ion fluid density. The neutral density is assumed to have a constant "neutral wind" velocity \( \mathbf{u} \). In the frame of reference with the neutral wind velocity, the neutrals act as a momentum sink for both the electron and ion fluids. The neutral density is also assumed incompressible so that its continuity equation can be neglected. We then consider only the ions and electrons in our equations.

In chapter 3 we have derived the general form of the momentum equation for plasmas. The complete momentum equation for species \( j \) in the ionosphere is

\[
n_j m_j \left( \frac{\partial}{\partial t} + \mathbf{v}_j \cdot \nabla \right) \mathbf{v}_j = q_j n_j \left( \mathbf{E} + \mathbf{v}_j \times \mathbf{B} \right) - \nabla P - \nu_{jn} n_j m_j (\mathbf{v}_j - \mathbf{u}) + n_j m_j \mathbf{g} \tag{5.1}
\]

where \( \mathbf{E} \) and \( \mathbf{B} \) are the electric and magnetic fields, \( m \) is the species mass, \( q \) is the charge of the species, \( P \) is the pressure proportional with temperature, \( \mathbf{g} \) is the gravity, and \( \nu_{jn} \) is the frequency of collision with the neutrals. The above momentum equation will be used to obtain the velocities for ions and electrons, but we will first make some simplifications.

Based on the relative importance of inertia versus ion-neutral collisions in the interchange instability, we can consider two separate regimes. The altitude range of about 200 - 400 km is called the collisional regime where the ion inertial effects can be neglected. As the altitude increases, ion inertial effects become more important because the ion-neutral collision frequency decreases due to reduction in the neutral density. Altitudes above 600 km where the ion inertial effects dominate the collisional effects is called the inertial regime. We will consider ionospheric instabilities occurring within the collisional regime, due to the simplicity of the model. We will, therefore, neglect the ion inertial effects. We have made this assumption because we are primarily interested in the numerical implementation of the spectral methods, and not so much in obtaining an exact model of ionospheric instabilities.

Electron inertial and electron-neutral collisions will also be neglected, since their effects are much smaller than those of the ions. These are valid assumptions for the F region. Similarly, we will neglect the finite temperature effects and gravitational acceleration in
our simulation model. The momentum equations for ions and electrons are then

\begin{align}
0 &= q_i n_i (E + v_i \times B) - \nu_{in} n_i m_i (v_i - u) \tag{5.2} \\
0 &= q_e n_e (E + v_e \times B) \tag{5.3}
\end{align}

From (5.3) the electrons simply move at the $E \times B$ velocity

\[ v_e = \frac{E \times B}{B^2} \tag{5.4} \]

as we have seen in section 3.7.3. To calculate the ion velocity, we assume that all frequencies are small compared to the ion gyrofrequencies $\Omega_i = q_i B / m_i$ from (3.19) and follow the derivation similar to Scales and Bernhardt [1991]. The ion velocity is expanded in a small parameter $\epsilon$ where $\epsilon \sim \nu_{in} / \Omega_i$, $\nu_i / L \Omega_i$, where $L$ is the length scale. Furthermore, we assume $v_i = v^{(0)}_i + v^{(1)}_i + \cdots$, where $v^{(n)}_i \sim \epsilon^n$ and $\epsilon \ll 1$. Substituting into (5.2) and ignoring higher order terms, we obtain

\begin{align}
\quad v^{(0)}_i &= \frac{E \times B}{B^2} \tag{5.5} \\
\quad v^{(1)}_i &= \frac{\nu_{in}}{B \Omega_i} (E + u \times B) \tag{5.6}
\end{align}

These are the zeroth and first order ion velocities perpendicular to $B$. Note that to the zeroth order the ions $E \times B$ drift as the electrons.

The total current density for a fluid consisting of ions and electrons is defined in (3.45) as

\[ J = n_e q_e v_e + n_i q_i v_i \tag{5.7} \]

We substitute (5.5) and (5.6) into (5.7) and use the condition for quasi-neutrality that $\nabla \cdot J = 0$ and $n_i \approx n_e$. We can then obtain a condition on the electrostatic potential $\phi$, where $E = -\nabla \phi$,

\[ \nabla \cdot \sigma_P \nabla \phi = \nabla \cdot [\sigma_P (E \times B)] \tag{5.8} \]

This is the potential equation. Here $\sigma_P = \frac{n_e \nu_{in}}{B n_i} \Omega_i$ is the Pedersen conductivity for the ions. Pedersen conductivity is the conductivity in the direction of the applied field, while Hall conductivity $\sigma_H$ is that perpendicular to the applied field. In our simulation model, $\sigma_H$ is negligible compared to $\sigma_P$. The $u \times B$ on the right-hand side of (5.8) may be interpreted as an applied ambient electric field. Conductivities and the applied electric field are all
in the plane perpendicular to the magnetic field, which is assumed to be constant in our simulation model.

The plasma continuity equations for the ions and electrons as given in (3.38) are

\[
\frac{\partial n_i}{\partial t} + \nabla \cdot (n_i \mathbf{v}_i) = 0 \tag{5.9}
\]

\[
\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \mathbf{v}_e) = 0 \tag{5.10}
\]

The potential equation (5.8) and the continuity equations (5.9) - (5.10) completely specify the physical plasma model of interest in three-dimension. In this work, only two dimensions perpendicular to \( \mathbf{B} \) are considered. We assume that there exists a uniform background magnetic field \( \mathbf{B} = B\hat{z} \), where \( \hat{z} \) is the field-aligned coordinate vector. To reduce the potential and continuity equations from three to two dimensions, we use the concept of field-line integration discussed by Bernhardt [1988]. Both the electrons and ions are magnetized and their \( \mathbf{E} \times \mathbf{B} \) drift is much larger than other drifts, so that the zeroth order velocity dominates. The ion continuity equation then may be eliminated by the quasi-neutrality condition. Our final 2-D field-integrated simulation model is

\[
\nabla \cdot N \nabla \phi = \nabla \cdot B[N\mathbf{u} \times \hat{z}] \tag{5.11}
\]

\[
\left( \frac{\partial}{\partial t} + \frac{\hat{z} \times \nabla \phi \cdot \nabla}{B} \right) N = 0 \tag{5.12}
\]

where \( N \) is proportional to the field-integrated Pedersen conductivity, which is also proportional to the field-line integrated values of the plasma density. These equations could also be obtained using the derivation found in Perkins et al. [1973].

The F region field-aligned irregularities will move without any change in their cross section if the neutral wind velocity and the \( \mathbf{E} \times \mathbf{B} \) drift are identical. When there is a relative flow between the neutral and plasma fluids, the cross section of an irregularity becomes distorted by internally generated electric fields. This process is called the \( \mathbf{E} \times \mathbf{B} \) instability or the gradient drift instability. Our simulation model lends itself to the occurrence of this type of interchange instability. \( \mathbf{E} \times \mathbf{B} \) develops in an inhomogeneous, weakly collisional, magnetized plasma that contains electric field that is perpendicular to both the magnetic field and the density gradient. It is usually associated with the structuring of ionospheric plasmas.
Figure 5.1: Physical mechanism of the $\mathbf{E} \times \mathbf{B}$ instability. (After Huba et al. [1983].)

The mechanism of the the 1-D $\mathbf{E} \times \mathbf{B}$ instability is explained in Huba et al. [1983] by considering a plasma such that $\mathbf{B} = B\hat{z}$, $\mathbf{E} = E\hat{y}$, $n = n(x)$ with $\frac{\partial n}{\partial x} > 0$ and $\nu_{en}/\Omega_e \ll \nu_{in}/\Omega_i \ll 1$, where $\nu_{an}$ is the neutral species $\alpha$ collision frequency and $\Omega_\alpha$ is the gyrofrequency of species $\alpha$. Assume a density perturbation $\delta n \sim \delta n \sin(k_y y)$ as shown in Figure 5.1. The presence of $\mathbf{E}$ causes the electrons and ions to $\mathbf{E} \times \mathbf{B}$ drift in the $x$-direction and gives rise to an ion Pedersen drift in the $y$-direction. This Pedersen drift induces a space charge perturbation electric field $\delta \mathbf{E}$, to which the plasma responds by drifting with a velocity $\delta \mathbf{v} = \delta \mathbf{E} \times \mathbf{B}$. $\delta \mathbf{v}$ causes the 'heavy' fluid perturbation to fall into the 'light' fluid (region I) and the 'light' fluid to rise in the the 'heavy' fluid (region II). This is the interchange instability phenomena. The $\mathbf{E} \times \mathbf{B}$ instability is manifested in our model in the following manner. Given a plasma density cloud in a region conducive to the development of the interchange instability. The $\mathbf{E} \times \mathbf{B}$ drift of the center of the plasma cloud counteracts the general $\mathbf{E} \times \mathbf{B}$ drift, and the center of the cloud has the slowest drift. The backside of the cloud catches up with the center, and a steep density gradients forms there [Zalesak, 1979]. Equatorial Spread-F (ESF) that was discussed in chapter 2 is a type of ionospheric irregularities that is caused by the $\mathbf{E} \times \mathbf{B}$ instability.

5.1.2 Theoretical model in spectral domain

The set of differential equations in real space that are to be solved numerically are (5.11) and (5.12). The neutral wind is assumed to be in the $x$-direction so that $\mathbf{u} = u_0 \hat{x}$. Furthermore, it is necessary to prevent numerical difficulties with steep gradients and physically unrealistic small-scale fluctuations. An artificial diffusion term will be included
on the right-hand side of the continuity equation (5.12) in the form of $D_\ast \nabla^2 N$, where $D_\ast$ is the artificial diffusion coefficient. This artificial diffusion is larger than the natural electron diffusion since mother nature provides different processes when gradients become unnaturally steep. We will use the smallest possible $D_\ast$ so that the dynamics of fluctuations with wavelength much larger than the grid spacing will be minimally affected.

The continuity equation is solved using the pseudospectral technique described in section 4.3. Using Fourier transforms, the spatial dependence in (5.12) is decomposed into its spectral representation. This reduces the PDE into the following ordinary differential equation

$$\frac{dN_k}{dt} + \frac{1}{B} \sum_{p+q=k} (p \times \hat{z}) \cdot q \phi_p N_q = -D_\ast k^2 N_k$$

(5.13)

where $k$ denotes the Fourier mode number. The coefficients are marched in time by using a combination of a leapfrog and leapfrog-trapezoidal time advances. To solve for the electric potential, we first perform the vector identity

$$\nabla \cdot (\psi A) = \psi \nabla \cdot A + A \cdot \nabla \psi$$

(5.14)

on the left hand side of (5.11), where $\psi$ and $A$ represent a scalar and a vector, respectively. The potential is then solved in Fourier space as

$$\sum_{p+q=k} N_p q^2 \phi_q + \sum_{p+q=k} p \phi_p \cdot q N_q = u_0 B i k_y N_k$$

(5.15)

using an iterative technique. The convolution terms in (5.13) and (5.15) are solved using the pseudospectral method with the nonlinear terms being fully dealiased. Sample runs of the continuity and potential equations will be presented in the next section.

Typical ionospheric parameters of altitudes 200 - 300 km are given in Table 5.1. In our simulation model, we will not be using these parameters. Instead, we restrict ourselves to using simulation units. This is because the emphasis of our work lies on the numerical techniques used in the study of plasma evolution.

5.1.3 Energy conservation

Energy is a quadratic quantity useful in understanding the physical processes being simulated. It is also useful in checking the integrity of the numerical simulation model.
Table 5.1: Typical parameter for ionospheric plasma 200 - 300 km. (After Bernhardt [1984])

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnetic field strength</td>
<td>B</td>
<td>$3.0 \times 10^{-5}$ T</td>
</tr>
<tr>
<td>Ion and neutral temperature</td>
<td>T</td>
<td>750 °K</td>
</tr>
<tr>
<td>Electron temperature</td>
<td>$T_e$</td>
<td>1000 °K</td>
</tr>
<tr>
<td>Neutral diffusion coefficient</td>
<td>D</td>
<td>$8.17 \times 10^{9}$ cm$^2$/s</td>
</tr>
<tr>
<td>Neutral wind velocity</td>
<td>$u_0$</td>
<td>200 m/s</td>
</tr>
<tr>
<td>Electron neutral collision frequency</td>
<td>$\nu_{en}$</td>
<td>$4.0 \times 10^{-8}$ s$^{-1}$</td>
</tr>
</tbody>
</table>

The integral of the square of density is a conserved quantity that is derived by multiplying (5.12) by $N$ and integrating over all space [Zarsham and Seyler, 1989]

$$\frac{d}{dt} E_{tot} = \frac{v_0^2}{2} \int \int \frac{d}{dt} N^2 dx dy = 0$$  \hspace{1cm} (5.16)

in the limit of $D_x \rightarrow 0$, where $N$ is the field-integrated Pedersen conductivity proportional to the plasma density $n$. By solving the equations in the Fourier space as opposed to real space (using finite-difference and other methods), the nonphysical numerical diffusion is less than that of the finite-difference method.

5.2 Two-dimensional simulation model

Program ESSPEC2* was developed to solve the set of equations (5.11) and (5.12). The algorithm of ESSPEC2, shown in Figure 5.2, is straightforward. The computation cycle consists of three major subroutines TIMADV, POTENTIAL, and CONTINUUK. In each cycle, TIMADV calls POTENTIAL to solve for $\phi_k$ in the potential equation. Subroutine POTENTIAL then calls CONTINUUK to calculate the right-hand side of (5.13). The plasma density is then updated in TIMADV using a combination of leapfrog and leapfrog-trapezoidal time advance algorithms and the program advances to the next time step. When total number of timesteps $nt$ is reached, the program ends. The choice of the number of timesteps is

* A complete listing of this program can be found in Appendix A.
based on several factors. We wish to run the simulation long enough to observe the physical processes under consideration. We maximize the step size while still obeying the Courant condition for numerical stability. All computation is done in the transform space. We only return to the real space to provide data for plotting.

We begin this section by discussing the fundamental tasks performed in ESSPEC2. These include performing DFTs, differentiation, Laplacians, and convolution using spectral methods developed in the last chapter. We then incorporate these operations to solve the potential and continuity equations, and then present our results. Emphasis is placed on the understanding of the basic algorithms of these operations. Readers should refer to Appendix A for more detailed features of ESSPEC2.

5.2.1 Fast Fourier Transform

Computation of forward and inverse discrete Fourier transforms (DFT) can be considered as the backbone for any numerical simulations that use spectral methods. The importance of DFT computation will become clear if we consider the number of transforms performed in each time step. A real-valued initial density is specified. This initial density is transformed into the Fourier space since all computations are done in the Fourier
spectral domain. Each computation of convolution requires four DFT operations (two for each array). A convolution is performed at least five times (about 15 times on the average) everytime the POTENTIAL subroutine is called.

As we can see, computation of DFTs is very important and has to be performed a number of times every time step. Hence, it is crucial that the algorithm for calculating the DFT we use be fast, accurate, and requires minimum amount of memory. We saw in the last chapter that DFTs can be computed with the Fast Fourier Transform (FFT). The FOURN and RLFT3 subroutines found in Press et al. [1993] are incorporated into the simulation program to compute FFTs. FOURN returns the N-dimension discrete Fourier transform. In our case, it replaces the input data with its two-dimensional discrete Fourier coefficients. FOURN is very versatile in that it can return the Fourier transform of both complex and real functions. The tradeoff, however, is the number of operation counts it requires. RLFT3, a driver subroutine for FOURN, allows us to use FOURN to simply treat the input data as a real, two-dimensional array. Hence, the operation count is halved. Special care must be used when performing any operation on the output array of RLFT3. The output spectrum of RLFT3 is in two complex arrays, one two-dimensional and the other one-dimensional [Press et al., 1993]. The two-dimensional array contains spectral components whose x-directed wavenumbers range from zero to just short of the Nyquist wavenumber \( \frac{2\pi n_x}{L_x} \), where \( n_x \) is the number of grid cells and \( L_x \) is the plasma length in the x-direction. The full range of positive and negative components of y-directed wavenumbers is stored in wrap-around order. The one-dimensional complex array stores only the single value in the x-direction that corresponds to the Nyquist wavenumbers, but all values in y-direction.

Having obtained the Fourier transformed values for any set of real input data, we are now in the position to perform spectral/pseudospectral operations on them.

5.2.2 Taking derivatives and Laplacians

The task of taking derivative and Laplacian in ESSPEC2 is done as follows. The discretized Fourier wavenumbers \( \kappa_x, \kappa_y \) and the corresponding discretized Laplacian coefficients \( K^2 \) are first calculated (see section 4.3.3). DERIVK multiplies each Fourier transformed value with the corresponding Fourier wavenumber, and then returns the Fourier transformed
$x$- or $y$-derivative of the grid quantity. A flag is placed to determine if the derivative is taken with respect to $x$ or $y$.

Evaluating the Laplacian in the Fourier domain is very similar to the computation of derivative. As in DERIVK, a flag is placed in LAPLACEK that determines which of the Laplacian or inverse Laplacian is to be computed. LAPLACEK calculates the Fourier transformed inverse Laplacian by dividing with $K^2$, or the Fourier transformed Laplacian by multiplying with $K^2$.

When performing derivatives or Laplacians, it is very important to put $\kappa$ and $K^2$ in the corresponding wrap-around form that is imposed by the output of RLFT3.

5.2.3 Convolution

As discussed in section 4.3.4, convolution is evaluated using the pseudospectral method [Orszag, 1971b]. Aliasing is minimized by grid shifting technique and truncation of Fourier modes that lie outside $.94K$ where $K = \frac{2\pi n}{2L}$ is half of the maximum number of Fourier modes Patterson and Orszag [1971]. Here we have used equal plasma length $L_x = L_y = L$. Grid shifting coefficients and the coefficients of the filter used in truncating specific Fourier modes are calculated in the beginning of the program.

Subroutine CONVOLVE performs the convolution calculation in ESSPEC2. When CONVOLVE is called, it first computes the grid shifted quantities for the input arrays. It then calculates the convolution with the convolution theorem, that is, returning to the real space to perform multiplication and then Fourier transform the product. Similar steps are taken for the nonshifted grid quantities. Back in the transform space, the shifted grid convolutions and the nonshifted grid convolutions are summed up according to (4.24). This is the final convolution result.

It is necessary to filter the input arrays that are to be convolved before engaging subroutine CONVOLVE. The coefficients of the filter can be modified to best fit the problem we wish to simulate. For example, we may change the filter coefficients to change the point when the filter should start to roll-off.

5.2.4 Solution of the continuity equation

We have briefly explained how we implement the continuity equation solver using the
pseudospectral method. We wish to find the solution to the continuity equation and present the results as generated by the continuity equation solver. Equation (5.12) is first generalized by referring to \( N \) as the density \( n \) and \( \frac{\hat{z} \cdot \nabla \hat{v}}{B} \) as the convecting velocity \( v \). The modified (5.12) states that the velocity field is incompressible and that \( n \) remains constant as it moves in time. In the spectral domain, this idea is described by the following ordinary differential equation

\[
\frac{dn_k}{dt} = - \sum_{p+q=k} v_p \cdot iq \cdot n_q
\]  

(5.17)

where \( k \) denotes the Fourier wavenumber. A diffusion term is not necessary in this case since no steep gradient should develop. Subroutine CONTINUUM calls DERIVK to calculate the \( iq \cdot n_q \) term in the transform space, and then calls CONVOLVE to calculate the convolution term. CONVOLVE then sums up all the convolution terms and returns the Fourier space values of the right-hand side of (5.17).

We will now present the results of a test to verify our algorithm. We follow closely Orszag's [1971a] classic simulation work of two-dimensional convection of a passive scalar by a uniform rotation velocity. The initial condition of density \( n \) is specified as a cone with base radius of 16 grid cells and height of 5 grid cells centered at (32,64), shown in Figure 5.3. This scalar field rotates uniformly about (64, 64) grid point with angular velocity \( \Omega \). The convecting velocity can then be expressed as \( v = -\Omega(y\hat{z} - x\hat{y}) \). We assign \( \Omega \) to \( \frac{\pi}{40} \) (positive \( \Omega \) for counter-clockwise rotation). The velocity field is plotted in Figure 5.4.

The main difference between our simulation parameters and the ones of Orszag [1971a] is the size of the simulation mesh. Orszag used \( 32 \times 32 \) grid. Hence, we have scaled our parameters against the size of the simulation box so that we can compare the results justly. Orszag's passive scalar convection result was shown earlier in Figure 4.4(a) of section 4.3 along with comparison to other numerical methods. It was clear from that Figure 4.4 that the spectral method is superior to other finite-differencing methods in that it produces a conic shape hardly modified from the original.

We show our results to the passive scalar convection problem in Figures 5.5 and 5.6. Figure 5.5 shows that after \( \frac{1}{4} \) revolution, the cone has not changed very much from its initial shape except that its peak has lost some sharpness, as we can gather from the contour lines values. After a full revolution, however, we see in Figure 5.6 that the cone has a trailing
Figure 5.3: Initial density $n$ (conical distribution) on $128 \times 128$ grid space.
Figure 5.4: Velocity field for uniform rotation.

Figure 5.5: Contours of density $n$ obtained after $\frac{1}{4}$ revolution.
wakes, in addition to a slight distortion on the shape of the cone. This really does not compare well to Orszag's [1971a] results. There are several explanations to what causes this phenomena. First, Orszag [1971a] used a Fourier spectral Galerkin method for his convolution calculation, while we compute the convolution using the pseudospectral method. As we have discussed in section 4.3.5, there are some trade-offs between spectral Galerkin and pseudospectral schemes. Although considerably simpler and faster, pseudospectral method sometimes cannot eliminate aliasing completely. We can eliminate more aliasing by filtering off more $k$ modes; nevertheless, this is done at the cost of losing more amplitude information thereby causing more smearing effects. Second, it is impossible to recover discontinuous functions such as our cone function after DFT and inverse DFT have been performed on them. To represent such discontinuities, we need an infinite bandwidth or an infinitely small grid size. The Fourier series representation fails to converge in the neighborhood of points of discontinuous function. The representation exhibits oscillatory behavior near points of discontinuity. This behavior is known as the Gibbs phenomenon.

To avoid the problems associated with discontinuous function, we define a new density using a gaussian distribution while keeping the rest of the parameters the same. The initial density is then

$$n(x, y) = \exp \left( -\frac{x^2 + y^2}{80} \right)$$

(5.18)

centered at (32,64) grid point shown in Figure 5.7. The base radius of this gaussian function is approximately equal to the cone base radius. After one full revolution, we see in Figure 5.8 that the results are much better. There is only a slight distortion on the shape of the gaussian density. From the contour lines we note that only small percentage ($\approx 2.5\%$) of amplitude information is lost. More importantly, there hardly exist any trailing wakes. Comparison to Figure 5.6(a) implies that any wakes that is in Figure 5.8(a) maybe attributed to the aliasing not eliminated by pseudospectral methods. Hence, we may conclude that the wakes we see in Figure 5.6 are caused by the Gibbs phenomenon due to the discontinuity in the cone distribution.

So far we have only demonstrated the solution of continuity equation in uniform rotation. However, some of the plasma evolution studies that interest us move densities across the simulation box. This is the translation motion. Thus, it is beneficial for us to solve (5.17)
(a) 3-D perspective plot of density after 1 revolution

(b) Contour plot of density after 1 revolution

Figure 5.6: Density \( n \) (conical distribution) after 1 revolution on 128 \( \times \) 128 grid space.
Figure 5.7: Initial density $n$ (gaussian distribution) on $128 \times 128$ grid space.
Figure 5.8: Density $n$ (gaussian distribution) after 1 revolution on 128 × 128 grid space.
Figure 5.9: Velocity field for translation in the positive x-direction.

when the convecting velocity is a constant in one direction and zero in the other. The new velocity field is shown in Figure 5.9 where $v_x$ is 2. That is, the gaussian structure moves 2 grid cells every 10 time steps (stepsize is 0.1). The initial density is found in Figure 5.7. After 800 time steps, the density structure has moved across 1.25 times the simulation box length. The results are given in the Figure 5.10. They are similar to the results obtained from the uniform rotation of the gaussian structure in that there is very slight distortion in the shape and an almost negligible trailing wake.

5.2.5 Solving the potential equation

We will briefly discuss the numerical solution of the potential equation in (5.11) by the spectral/pseudospectral methods. We can rewrite the potential equation in a general form

$$\nabla A \nabla \phi = B$$

(5.19)

where $A$ and $B$ are space and time dependent coefficients. We perform the vector identity stated in (5.14) on (5.19) and find the corresponding potential equation in the Fourier
(a) Contour plot of density after 400 time steps

(b) Contour plot of density after 800 time steps
DENSITY
T = 80.00

(c) 3-D perspective plot of density after 800 time steps

Figure 5.10: Density $n$ (gaussian distribution) of translation in the positive x-direction.
domain
\[ \sum_{p+q=k} A_p q^2 \phi_q + \sum_{p+q=k} p \phi_p \cdot q A_q = -B_k \]  
(5.20)

Solving for \( \phi_k \) in 2-D has to be done carefully. We first move the second left hand side term of (5.20) to the right hand side. Next, we perform the inverse DFT and divide the resulting equation by \( A \) to isolate the \( \nabla^2 \phi \) term. We return to the Fourier domain and perform the inverse Laplacian. We then obtain
\[ \phi_k = \frac{1}{k^2} \sum_{p+q=k} \left( \frac{1}{A} \right)_p \left[ \sum_{m+n=q} m \cdot n \phi_m A_n - B_q \right] \]  
(5.21)

where the \( \left( \frac{1}{A} \right)_p \) term is obtained from the division performed in the real space. Note that the \( \phi \) term appears on both sides of the equation. There are several methods that can be used to calculate \( \phi_k \) in (5.21). The simplest method is to perform iterations. An initial guess of \( \phi \) on the right-hand side of (5.21) is first specified. We then filter out the Fourier modes that contain mostly noise. Then we iterate on \( \phi_k \) until the convergence criterion on the maximum allowable error is satisfied. In every iteration \( n \) we look for the maximum value of \( \phi_k \) and compare it to the maximum value of \( \phi_k \) of the previous iteration. The following criterion
\[ \left| \frac{\phi_k^{n+1}(\text{max}) - \phi_k^n(\text{max})}{\phi_k^{n+1}(\text{max})} \right| < \text{max error} \]  
(5.22)

is used to test the convergence of \( \phi_k \). On average, \( \phi_k \) converges after 4 or 5 iterations.

In our simulation program ESSPEC2, potential equation solver is implemented as follows. Given \( A \) and \( B \), subroutine POTENTIAL calls LAPLACEK, DERIVK and CONVOLVE to perform the iterations. \( \phi_k \) is returned.

A special case of (5.19) is the Poisson’s equation obtained by setting \( A = 1 \) and \( B = -\frac{\rho}{\epsilon_0} \). Equation (5.21) simply becomes
\[ \phi_k = \frac{\rho_k}{k^2} \]  
(5.23)

where \( k = k_x \hat{x} + k_y \hat{y} \) and \( k^2 = k_x^2 + k_y^2 \) in two-dimensions, and \( \epsilon_0 \) has been set equal to 1. We will use this special case to test our potential equation solver.

An analytic function of \( \rho \) is specified using a product of sines and cosines in the form
\[ \rho(x, y) = \cos \left( \frac{2\pi}{L} x \right) \sin \left( \frac{2\pi}{L} y \right) \]  
(5.24)
Figure 5.11: Density contours for $\rho(x, y) = \cos 2 \left( \frac{2\pi}{L} x \right) \sin 3 \left( \frac{2\pi}{L} y \right)$. 

on which we have imposed the following conditions for doubly periodic boundary and charge neutrality conditions [Birdsall and Langdon, 1991]:

- $\rho(x = 0, y) = \rho(x = L, y)$
- $\rho(x, y = 0) = \rho(x, y = L)$
- $\int_A \rho \ dx \ dy = 0$

where $L$ is the length of the system. We equate $L$ to 128, the dimension of our simulation box. The contour plot of $\rho$ is shown in Figure 5.11. From $\rho$, we can derive an analytical solution to $\phi$

$$\phi(x, y) = \frac{1}{13} \left( \frac{L}{2\pi} \right)^2 \cos 2 \left( \frac{2\pi}{L} x \right) \sin 3 \left( \frac{2\pi}{L} y \right)$$

that is given in Figure 5.12. We compare the derived analytical potential and the computed potential function $\phi$ depicted in Figure 5.13. As we can see, Figures 5.12 and 5.13 are practically identical. The contour lines differ only by very small numerical differences.

We can, furthermore, verify the integrity of subroutine DERIVK. Having obtained the potential, we take the second derivative of $\phi$ with respect to $x$ and add it to the second
Figure 5.12: $\phi = \frac{1}{13} \left( \frac{L}{\ell} \right)^2 \cos 2 \left( \frac{2\pi}{L} x \right) \sin 3 \left( \frac{2\pi}{L} y \right)$ as solved analytically from $\rho$.

Figure 5.13: $\phi$ as computed by subroutine POTENTIAL.
Figure 5.14: Laplacian of $\phi$, should be identical to $-\rho$.

derivative of $\phi$ with respect to $y$. This is essentially the Laplacian of $\phi$ and is accomplished by calling DERIVK twice for each $x$- and $y$-derivatives. The results should be identical to the negative of the original $\rho(x, y)$, which is shown in Figure 5.14.

5.2.6 Diagnostics

A number of diagnostics is incorporated into program ESSPEC2 to interpret the simulation results.

- Density plots (contour plots or surface plots)
- Velocity vector plots
- Potential and electric field plots
- Total energy versus time
- Animation of the density profiles

The first three type of plots are produced at a regular interval, while the last two are generated at the end of the simulation run. As we discuss the results of our simulation in the next section, selected plots will be presented.
5.3 Simulation Results

In the last sections we have derived the theoretical model that governs the nonlinear motion of plasma clouds as they evolve in the ionosphere. We separately tested the potential equation and the continuity equation solvers, and as we have seen earlier, the results produced by these solvers were satisfactory. We then integrate these solvers along with the time advancing technique and other subroutines that we have developed into our 2-D fluid electrostatic program. Appropriate initial and boundary conditions are applied to closely model specific characteristics of plasma evolution in the F region.

In this section we discuss the results of our simulation runs. We first present the result of plasma evolution using slab model since it has the simplest geometry. Next, the evolution of a density enhancement evolution will be discussed, followed by the density depletion case. Relevant simulation work and experimental results will be referenced when appropriate. We consider applications to ionospheric phenomena, i.e., barium cloud striation, irregularities caused by electron attachment materials, and F region natural irregularities.

5.3.1 Slab model evolution

We first consider the slab model of plasma evolution since it is the simplest plasma model. The geometry of the slab model can be explained as follows. Consider a plasma density cloud on the xy-plane with the constant magnetic field lines aligned along the z-axis as shown in Figure 5.15. A neutral wind $u_0$ is present along the x-direction. The plasma of interest has been divided into thin layers or “slabs” of plasma perpendicular to the magnetic field.

We consider a slab with initial density irregularities. As the initial conditions, a collection of randomly excited large-amplitude perturbations is superimposed on the plasma density. The initial state of the slab density is given below

$$N(x, y) = 0.01 \sum_i \sin \left( \frac{2\pi m_i}{L} x + p_{zi} \right) \sin \left( \frac{2\pi n_i}{L} y + p_{yi} \right)$$  (5.26)

where $0 < m_i, n_i \leq 8$ and $0 < p_{zi}, p_{yi} < 2\pi$ represent randomly generated frequencies and phase shifts, respectively. The initial density profile is shown in Figure 5.16. Due to the slab geometry, these irregularities are embedded in a constant density background so that they are not driven by a background density gradient. We can then specify the neutral wind
Figure 5.15: Slab model of plasma. (Modified Figure 2 from Zalesak et al. [1985].)

PEDERSEN CONDUCTIVITY

\[ T = 0.000 \]

Figure 5.16: Initial density profile of the slab model of plasma.
velocity of 75 and an artificial diffusion coefficient of 0.4. The time step size is chosen to be 0.0125 to satisfy the Courant condition. These values are given in the simulation units. As we shall see later, $u_0 = 75$ is much larger than the velocity value used in the plasma cloud model, given identical artificial diffusion constant. We may use larger value for the velocity because using the slab model minimize the numerical difficulties that may arise due to steep density gradients.

The late-time profile is shown in Figure 5.17. The initially isotropic density profile in Figure 5.16 has evolved to an anisotropic one. The density forms long, finger-like structures or striations along the neutral wind direction. This is a manifestation of the gradient drift instability. This late-time density profile can be qualitatively described by the presence of quasi-periodic structures in the $y$-direction (parallel to the ambient electric field direction or $u \times B$ direction) and propagating shocklike structures in the $x$-direction.

Our results are similar to the simulation results of nondriven irregularities performed by Zargham and Seyler [1987]. We have considered the development of the $E \times B$ instability while Zargham and Seyler considered the generalized Rayleigh-Taylor (GRT) instability. We
have also used a slightly different geometry set up for our slab model and slightly modified potential and continuity equations used to describe the nonlinear plasma evolution. In general, this type of slab model evolution can be used to study large-amplitude density variations near and above the F peak.

5.3.2 Density enhancement evolution

Density enhancement evolution has been extensively studied [Zabusky et al., 1973; Chaturvedi and Ossakow, 1977; Zargham and Seyler, 1989; and many others] to understand the evolution of artificial plasma clouds created by the release of neutral barium. In our simulation we follow closely the work of Zabusky et al. [1973]. The theoretical model we derived earlier in this chapter is essentially the same. Using the spectral/pseudospectral methods discussed in chapter 4, we attempt to reproduce their results that were obtained by finite difference methods and show that our results are valid.

Plasma cloud model is a more realistic model for the plasma evolution than the slab model. The initial state of the 2-D plasma cloud is composed of a gaussian-like distribution

\[ N(x, y) = 1 + 4 \exp \left( -\frac{p_0}{(2d_0)^6} \right) \]  

(5.27)

where \( N \) is the field integrated Pedersen conductivity and \( d_0 \) is a parameter determining the plasma cloud base radius. The background conductivity is normalized to 1. We will use the term 'Pedersen conductivity' and the term 'density' interchangeably. The initial conductivity has a weak trigonometric perturbation in the following form

\[ p_0(x, y) = [(x^2 + y^2)(1 - A \cos(B\theta))]^3 \]  

(5.28)

where \( \theta = \arctan(x, y) \), \( A \) indicates the strength of the perturbation (0 for no perturbation), and \( B \) specifies the frequency of the sinusoidal perturbation. We let \( A = 0.015 \) and \( B = 12 \). The strength of the initial perturbation is proportional to how fast striations develop. Since this perturbation is artificial in nature, it is large enough to just facilitate the initial striation without overshadowing the more realistic processes that occur during the nonlinear evolution.

Next, we specify the neutral wind velocity \( u_0 \) of 2.5 in the positive x-direction, and the artificial diffusion constant of 0.1. Note that for equal value of diffusion constant, the
neutral velocity in the plasma cloud model is much smaller than that of the slab model. This is due to a steep density gradient that exists in the plasma cloud model. We may increase the neutral wind velocity or decrease the diffusion constant if the height of the plasma cloud is lower, since the density gradient decreases. Similarly, increasing the base radius of the cloud while keeping its height constant reduces the density gradient. The step size is 0.05 to satisfy the Courant condition.

The isodensity contours for the initially circular plasma cloud enhancement with weak initial perturbation is given in Figure 5.18. Note that the cloud is moving from left to right along with the neutral wind velocity. We assume the release occurs at the center of the simulation box. Doubly periodic condition is imposed so that the density that leaves the simulation box from one side will enter from the opposite side. The perturbation is weak, though evident at $T = 0$. At $T = 20$ the backside oscillations have vanished, whereas the frontside has steepened. The cloud begins to elongate and bifurcate as seen at $T = 40$. The late-time profile shows that the cloud has completely bifurcated and the striation sheets begin to fission.

Our results are in agreement with those obtained by Zalesak et al. [1973] using finite-difference method. The qualitative features of the density profile at late-times can be described as follows. The fissioned rods are elongated with steep and dense frontside. The backside is shallow with a long tail. We can see that steepening is also present in the inner side of the rods. If the frontside broadens sufficiently, it will become unstable to a mode similar to the mode that existed initially. If there is sufficient plasma in the striation, another bifurcation might occur.

Surface plots of the early-time and late-time density profiles are shown in Figure 5.19. Note that where steepenings are present, there exist peaks that can be associated with the sharp discontinuities that give rise to the Gibbs phenomena. If the steepening continues, the simulation model may eventually become numerically unstable. Another aspect that may contribute to the numerical instability of our model is the interference of the tail of the cloud with the center of the cloud. As depicted in Figure 5.19(b), it is clear that the tail of the cloud will eventually become long enough. A solution to this particular problem is to make a larger simulation box.
PEDERSEN CONDUCTIVITY

\( T = 0.000 \)

(a) \( T = 0 \)

PEDERSEN CONDUCTIVITY

\( T = 20.000 \)

(b) \( T = 20 \)
Figure 5.18: Isodensity contours for the plasma cloud enhancement evolution.
Figure 5.19: Surface plots of the initial and late-time density profiles of the density enhancement problem.
Figure 5.20: Energy history plot for density enhancement evolution.

Finally, we check the quality of our numerical solution by considering the energy history plot, shown in Figure 5.20. As was discussed earlier in (5.16), the energy is a conserved quantity. The energy in shown to be fairly constant (variation of less than 0.1 or 0.00004%) throughout the simulation run. Thus, good numerical solutions are indicated.

Our computational results may be used to show features of field-aligned observations of barium releases. Rosenberg [1971] observes three striations that develop from the steepening of the gradient facing the neutral cloud. This phenomena was observed 500 seconds after the release of a barium ion cloud at 194 km (see Figure 2.7).

5.3.3 Density depletion evolution

The chemicals that are used in the release of electron attachment materials and the chemical reactions involved were discussed earlier in section 2.4.6. The problem of electron attachment materials is not as well studied as the barium release counterpart. Some of nonlinear simulations of this type of release are Bernhardt [1988] and Scales and Bernhardt [1991]. The electrostatic fluid simulation study of Bernhardt [1988] considered the ionospheric irregularities subjected to the influence of a neutral wind generated electric field.
It showed that at early times the irregularity $E \times B$ drifts in the direction of the neutral wind. No structuring is presence due to the neutralization chemistry. At late times, after all the neutral attachment materials have been used up and depletion has formed, the irregularity drifts in the direction opposite to the neutral wind at much higher speed. The depletion steepens and structures due to the $E \times B$ interchange instability. Scales and Bernhardt [1991] incorporated ion inertial effects which are essential to correctly model the irregularities at high altitude.

In our work we consider a model similar to Bernhardt [1988] where the ion inertial effects are not important and the ionospheric irregularity is subject to the influence of a neutral wind generated electric field. The above-mentioned simulation had used finite-difference method to numerically solve the set of nonlinear equations. We will use the spectral/pseudospectral treatment in our simulation, as discussed earlier. Our simulation will model the late-time behavior, after all the attachment materials have been used up and density depletion has formed.

Similar set up to the density enhancement case is used. The initial state of the 2-D plasma cloud is composed of the same gaussian-like distribution

$$N(x, y) = 1 - 0.55 \exp \left[ -\frac{p_0}{(2d_0)^6} \right]$$

where the plasma cloud base radius and the perturbation are identical to the ones used earlier in the density enhancement model. We decrease the maximum height of the Gaussian structure from 5 to 0.55 units. We specify the neutral wind velocity $u_0$ of 2.4 in the positive $x$-direction, and the artificial diffusion constant of 0.15. Note that we have chosen a slightly smaller value for $u_0$ and a larger value for diffusion constant to take care of the faster drifts in the depletion case.

The development of isodensity contours for the initially circular plasma cloud depletion with weak initial perturbation is given in Figure 5.21. We assume the release occurs at the right side of the simulation box. Again, doubly periodic condition is imposed so that the density that leaves the simulation box from one side will enter from the opposite side. Note that the neutral wind is from left to right in the simulation box and the cloud is drifting opposite of the neutral wind. At $T = 20$ on the upwind side has begun to steepen. The cloud begins to elongate and bifurcate as seen at $T = 40$. The late-time profile shows that
the depletion has completely bifurcated and sheared into two pieces. The striations consist of a set of long straight structures parallel to the direction of the neutral wind. The basic results in Figure 5.21 are in agreement with the work performed by Bernhardt [1988] and the work on the collisional limit of the interchange instability by Scales and Bernhardt [1991].

The surface plot of the late-time density profile is given in Figure 5.22. This figure should actually be rotated 90 degrees around the center of the simulation box. As before, where sharp discontinuities occur such as the tip of the steepened rods, a series of peaks that can be associated with the Gibbs phenomena is present.

We check the quality of our numerical solution by considering the energy history plot, shown in Figure 5.23. As for the enhancement case, the total energy is shown to be fairly constant (variation of less than 0.1 or 0.0006%) throughout the simulation run.

5.3.4 Comparison of density enhancement and density depletion evolutions

There are noted differences of the results between the density enhancement and density depletion cases. We can best compare the differences by running two separate simulations having two different initial densities while keeping the rest of the parameters identical. The initial conductivity of the enhancement model ranges from 1 to 1.55, while the initial conductivity of the depletion model ranges from 0.45 to 1. Both are located at the center of the simulation box.

Three sets of plots comparing the differences of the plasma enhancement and depletion models are found in Figures 5.24, 5.25, and 5.26. We have noted earlier that the enhancement drifts in the same direction with the neutral wind, while the plasma cavity drifts much faster in the opposite direction. This is because the Pedersen conductivity depletion results at late times create a polarization field such that the depletion \( E \times B \) drifts in the direction opposite to the neutral wind [Bernhardt, 1988]. We first compare the associated Pedersen conductivities at \( T = 30 \). The enhancement model has just begun to steepen, while the depletion model has actually bifurcated. The velocity of the drift is proportional to the electric field which is the divergence of \( \phi \). We may make a qualitative analysis on the drift velocity by examining the electrostatic potential plots in Figure 5.25. The potential for the enhancement case has an opposite sign and is smaller than that of the depletion
PEDERSEN CONDUCTIVITY

(a) $T = 0$

(b) $T = 20$
Figure 5.21: Isodensity contours for the plasma cloud depletion evolution.
Figure 5.22: Surface plots of the late-time density profile of the density depletion problem.

Figure 5.23: Energy history plot for density depletion evolution.
(a) enhancement model

(b) depletion model

Figure 5.24: Comparison of density profiles at $T = 30$ of the enhancement and depletion models.
case. Hence, the drift velocity of the enhancement model is also smaller and in the opposite direction.

A comparison of the animation of the density profiles\footnote{Available from Dr. W. A. Scales EE Dept., 615 Whittemore Hall, VPI&SU} provide more insights on how the development of striations in the enhancement and depletion cases differ. This strengthens our argument that the physical explanation of these types of irregularities may be vastly different.

A more quantitative analysis can be used to predict the plasma depletion structures caused by the $E \times B$ interchange instability. We define the collisional growth rate $\gamma_c$ [Scales and Bernhardt, 1991]

\[
\gamma_c = - \left( u_0 - \frac{E}{B} \right) \frac{1}{N} \frac{\partial N}{\partial x}
\]  

(5.30)

The right hand side of (5.30) represents the growth rate for the gradient drift instability which is dependent on the speed of the plasma relative to the background neutrals ($u_0 - \frac{E}{B}$), and the inverse gradient scale length $N^{-1} \frac{\partial N}{\partial x}$. In our simulation model we assume that the inverse gradient scale lengths are the same for the depletion and the enhancement models.

We calculate the ratio of $\gamma_c$ of the depletion with respect to that of the enhancement model and find the value to be 7.37. Since the growth rate for the depletion case is much larger than that of the enhancement case, we expect that the density depletion to steepen and bifurcate much faster. This can be observed clearly in Figure 5.26. The late-time profile of the enhancement model has only steepened, while the depletion model has bifurcated and striated.

In conclusion, we have shown that spectral/pseudospectral methods used in our 2-D fluid electrostatic model perform as well in solving nonlinear PDEs. Our results agree with the numerical results obtained by using other numerical methods, and also with the observation results.
Figure 5.25: Comparison of the equipotential profiles at $T = 30$ of the enhancement and depletion models.
Figure 5.26: Comparison of density profiles at late-times ($T = 60$) of the enhancement and depletion models.
6 Conclusions and recommendations

Two-dimensional electrostatic fluid plasma simulation using Fourier spectral and pseudospectral methods has been developed and performed to study the evolution of a plasma cloud in the ionosphere. We have considered the $\mathbf{E} \times \mathbf{B}$ interchange instability occurring within the collisional regime, due to the simplicity of the model. We have made this assumption because in this research work we are primarily interested in the numerical implementation of the spectral methods, rather than detailed ionospheric model.

Although the plasma model in not new, to our knowledge, the type of numerical simulation we developed has not been done before. We use the simple plasma model as derived by Perkins et al. [1973] and implemented numerically by Zabusky, [1973]. Since then a number of numerical simulation on plasma evolution have been performed [Zalesak et al., 1985; Mitchell et al., 1985; Scales and Bernhardt, 1991; and many others]. All the simulation work mentioned above use finite difference methods for their spatial discretization. Solving nonlinear partial differential equations as prescribed by the plasma model using finite differencing is not a trivial task. A number of algorithms have been designed to solve this problem. For example, FISHPAK ver. 3 [Adams et al., 1979] was developed at the National Center for Atmospheric Research to solve the standard five-point finite difference approximation. We believe our algorithm to be shorter and more compact than many widely used potential solvers. We have also developed an algorithm using spectral/pseudospectral methods to solve the fluid continuity equation, one of the hardest equations to be solved numerically. We found that our algorithm performed satisfactory. As we have discussed in chapter 4, spectral and pseudospectral methods have several advantages over finite difference and finite element techniques with the main advantages being their accuracy and their ability to handle nonlinear terms.

Solving the nonlinear plasma model equations using spectral and pseudospectral methods have been considered by Zargham and Seyler [1987 and 1989]. In their 1987 paper, they considered a similar model that is solved using spectral methods. This model is applied to a slab geometry with random perturbation. The slab model is much simpler in that the numerical instability due steep density gradient is nonexistent. In their 1989 paper, Zargham
and Seyler discussed solution to the plasma model using spectral methods as applied to the cylindrical plasma bubble.

The novelty of our work lies in that we apply spectral and pseudospectral methods in the simulation model of a plasma cloud, an initial density profile which is a much more realistic than the slab geometry. We also developed an algorithm to solve the potential in the potential equation using iterative method that is stable and robust. As discussed in chapter 5, we have successfully applied these numerical methods to this type of problem. We consider the evolution of an initially enhanced plasma density and an initially depleted plasma density. Some important conclusions are noted. First, the electric field causes the cloud to drift with an $E \times B$ velocity. Second, using the same level of perturbation, the density depletion case bifurcates much faster than the enhancement case. This idea has not been considered before. We find the ratio of the growth rate of the depletion to the enhancement to be $7.37 \gg 1$. However, at this point we are not sure what causes the depletion potential or electric field to be much greater than that of the enhancement.

Future efforts should be directed to make the simulation model more realistic. Ways of accomplishing this goal are suggested below:

- Extend the 2-D model to the corresponding 3-D simulation model.
  The FFT subroutine is already generalized to three-dimensions. Minor changes are needed to include the third dimension in the convolution, derivatives, Laplacians and the potential and continuity equation solvers.

- Include the electron attachment chemistry or photoionization chemistry depending on the type of releases.

- Include the ion-inertial effects since at high altitudes the ion-neutral collision frequency becomes negligibly small.

- Provide a more detailed explanation and comparison between density cloud enhancement and depletion in terms of their respective growth rate and direction of drift.
Appendix A

ESSPEC2 program listing

This is the FORTRAN program written to implement the numerical solution of the simulation model discussed in chapter 5. It solves the set of electrostatic fluid equations using the spectral methods. The run-time of ESSPEC2 program is about 4 cpu hours using the IBM RISC 6000 workstation for 1200 time steps. Steeper initial density requires more iterations in the potential equation solver, hence it takes longer time to complete.

The listing of ESSPEC2 is as follows

```
******************************************************************************
* * program esspec2 -- 2D electrostatic code with spectral and pseudo- *
*   spectral methods
* *
*   Leapfrog and leapfrog trapezoidal methods are used for time advancing, *
*   and iterations are used to solve for the potential in the potential *
*   equation
*
******************************************************************************

common /efluid/ bzi,ifts
real ne(129,129)
real l

character*32 edens
logical data_ex

inquire (FILE = 'density.image', EXIST = data_ex)
if(data_ex .eqv. .false.) goto 300
open (UNIT = 10, FILE = 'density.image', STATUS = 'OLD')
close(UNIT = 10, STATUS='DELETE')
300 open (UNIT = 20, FILE = 'density.image', STATUS = 'NEW')

t0=mclock()
call OPNGKS

c  l = length of the system
c  ng = number of grid points
c  nx = number of grid points in x-direction
c  ny = number of grid points in y-direction
c  nt = number of time steps
c  dt = time step size
c  bz = constant magnetic field B in the z-direction
c  u0 = neutral wind velocity in positive x-direction
c  ne = cloud density (integrated Pedersen conductivity)
```
twopi=8.*atan(1.)

l=128.
dt=0.05
ng=128
nt=1200

c
ng1=ng+1
ng2=ng/2
nx=ng
ny=ng
bz=1.
bzi=1./bz
u0=2.5
uxbz=u0*bz

c
r0=radius of cloud, (x0,y0)=center coord

c
x0=64.
y0=64.
d0=10.
do 10 j=1,ny
   ji=j-1
   y1=j1-y0
   do 10 i=1,nx
      ii=i-1
      x1=ii-x0
      theta=atan2(x1,y1)
      rsq=(x1)**2+(y1)**2
      rsq=rsq*(1.+0.015*cos(12.*theta-twopi/2))
      rsq=rsq*rsq*rsq
      ne(i,j)=1. + 4*exp(-rsq/((2*d0)**6))
c
   10 continue

do 110 j=2,ny
   ne(1:29,j)= ne(1,j)
110 continue

do 120 i=2,nx
   ne(i,1:29)= ne(i,1)
120 continue

ne(1,129)= ne(1,1)
ne(129,1)= ne(1,1)
ne(129,129)=ne(1,1)

write(edens,500)
500 format(’,PEDERSEN CONDUCTIVITY’).
call plotting(ne,1,ng1,edens,0.,1)
call plotting(ne,1,ng1,edens,0.,2)
call fldinit(ng,1)
call timadv(ne,ux,bz,l,l,1,.,ng,ng,1,nt,dt)

close(20)
call CLSGKS
t=(mclock()-t0)/100.

if (t.lt.3600.) then
  write(6,*), 'The total CPU time was ',t,' seconds'
else
  th=t/3600.
  write(6,*), 'The total CPU time was ',t,' seconds, or ',
  , th,' hours'
endif

stop
end

SUBROUTINE timadv(ne,ux,bz,lx,ly,lz,nx,ny,nz,nt,dt)

use 8 leap frog trapezoidal sequences followed by 1 leapfrog sequence

to advance the continuity equation

common /efluid/ bzi,ifts
common /error/ errmax
complex nek(64,128), neq(128)
complex nesk(64,128),nesq(128)
complex exk(64,128), exq(128), eyk (64,128), eyq (129)
complex phi(k(64,128),phiq(126),phisk(64,128),phisq(128)
complex rk(64,128),rq(128),rk1(64,128),rq1(128)
complex rkv(64,128),rkvq(128)
complex nepx(64,128), neqx(128), nemk(64,128), nemq(128)
real ne(129,129), nes(128,128)
real phi(128,129), phis(128,128)
real ex(129,129), ey(129,129), exs(128,128), eys(128,128)
real lx,ly,lz,1
real totheneg(80),ndens
character*32 edens,exf,eyf,pot

equivalence(nes,nesk)
equivalence(exs,exk)
equivalence(eys,eyk)
equivalence(phisk,phisk)

scale = nx*ny*nz
scale = 2./scale

nt1=nt+1
l=lx

ermax = 2.5e-4
errmax = 5.e-5
\[ nx2 = nx / 2 \]
\[ ng = nx \]
\[ ng1 = ng + 1 \]
\[ iplot = 1 \]
\[ ifreq = 20 \]
\[ endtime = dt * nt \]

555  \textbf{format('time = ',f7.3,' end time = ',f7.3,' no of iteration . = ',i3)}

\textbf{do 10 j=1,ny}
  \textbf{do 10 i=1,nx}
10  \textbf{nes(i,j) = ne(i,j)}
  \textbf{call rlft3(nes,nesq,nx,ny,nz,1)}
  \textbf{do 15 j=1,ny}
    \textbf{neq(j) = nesq(j)}
    \textbf{do 15 i=1,nx2}
15  \textbf{nek(i,j) = nesk(i,j)}
  \textbf{do 30 j=1,ny}
    \textbf{nemq(j) = neq(j)}
    \textbf{do 30 i=1,nx2}
30  \textbf{nemk(i,j) = nek(i,j)}
\textbf{continue}

\textbf{call potential(nek,neq,uxbz, . exk,exq,eyk,eyq,phik,phiq,rk,rq,nx,ny,nz,it)}

c  \textbf{BEGIN TIME LOOP}

c  \textbf{iic=0}

\textbf{do 1 icount = 1, nt}
  \textbf{ndens=0.}

  \textbf{if(mod(icount,8).eq.0)then}
    \textbf{ileapfr=1}
  \textbf{else}
    \textbf{ileapfr=2}
  \textbf{endif}

c  \textbf{leap frog trapezoidal}

c  \textbf{if (ileapfr.eq.1) then}

    \textbf{do 40 j=1,ny}
      \textbf{nepq(j) = nemq(j) + 2*dt*rq(j)}
      \textbf{do 40 i=1,nx2}
        \textbf{nepk(i,j) = nemk(i,j) + 2*dt*rk(i,j)}
40    \textbf{continue}

  \textbf{call potential(nepk,nepq,uxbz, . exk,exq,eyk,eyq,phik,phiq,rk1,rq1,nx,ny,nz,it)}

113
do 50 j=1,ny
   rqav(j)=0.5*(rq(j)+rq1(j))
do 50 i=1,nx2
   rkav(i,j)=0.5*(rk(i,j)+rk1(i,j))
50 continue

do 60 j=1,ny
   nepq(j)=neq(j)+dt*rqav(j)
   nemq(j)=neq(j)
   neq(j)=nepq(j)
do 60 i=1,nx2
   nepk(i,j)=nek(i,j)+dt*rkav(i,j)
   nemk(i,j)=nek(i,j)
   nek(i,j)=nepk(i,j)
60 continue

endif

c

straight leap frog
c

if(ileapfr.eq.2) then

do 70 j=1,ny
   nepq(j)=nemq(j)+2*dt*rq(j)
   nemq(j)=neq(j)
   neq(j)=nepq(j)
do 70 i=1,nx2
   nepk(i,j)=nemk(i,j)+2*dt*rk(i,j)
   nemk(i,j)=nek(i,j)
   nek(i,j)=nepk(i,j)
70 continue

endif

   call potential(nek,neq,uxbz,
     . exk,exq,eyk,eyq,phiq,phiq,rk,rq,nx,ny,nz,it)

time=dt*icount

c

PLOTTING
c

if (mod(icount,ifreq).eq.0) then
   iic=iic+1
do 105 j=1,ny
   nesq(j) = neq(j)
   phisq(j) = phiq(j)
do 105 i=1,nx2
   nesk(i,j) = nek(i,j)
   phisk(i,j) = phiq(i,j)
105 continue

   call rlft3( nes, nesq, nx,ny,nz,-1)
   call rlft3( exs, exq, nx,ny,nz,-1)
   call rlft3( eys, eyq, nx,ny,nz,-1)
   call rlft3( phis, phisq, nx,ny,nz,-1)
DO 110 J=1,NY
   DO 110 I=1,NX
      NE(I,J)= NES(I,J)*SCALE
      EX(I,J)= EXS(I,J)*SCALE
      EY(I,J)= EYS(I,J)*SCALE
      PHI(I,J)= PHIS(I,J)*SCALE
   CONTINUE
110  CONTINUE

DO 120 J=2,NY
   NE(129,J)= NE(1,J)
   EX(129,J)= EX(1,J)
   EY(129,J)= EY(1,J)
   PHI(129,J)= PHI(1,J)
120  CONTINUE

DO 130 I=2,NX
   NE(I,129)= NE(I,1)
   EX(I,129)= EX(I,1)
   EY(I,129)= EY(I,1)
   PHI(I,129)= PHI(I,1)
130  CONTINUE

NE(1,129)= NE(1,1)
RE(129,1)= NE(1,1)
NE(129,129)= NE(1,1)

EX(1,129)= EX(1,1)
EX(129,1)= EX(1,1)
EX(129,129)= EX(1,1)
EY(1,129)= EY(1,1)
EY(129,1)= EY(1,1)
EY(129,129)= EY(1,1)

PHI(1,129)= PHI(1,1)
PHI(129,1)= PHI(1,1)
PHI(129,129)= PHI(1,1)

WRITE(EDENS,500)
500  FORMAT (' PEDESEN CONDUCTIVITY ')
 WRITE(EXF,600)
600  FORMAT (' EX FIELD ')
 WRITE(EYF,700)
700  FORMAT (' EY FIELD ')
 WRITE(POT,800)
800  FORMAT (' POTENTIAL ')

CALL PLOTTING(NE,1,NG1,EDENS,TIME,IPLOT)
CALL PLOTTING(EX,1,NG1,EXF,TIME,IPLOT)
CALL PLOTTING(EY,1,NG1,EYF,TIME,IPLOT)
CALL PLOTTING(PHI,1,NG1,POT,TIME,IPLOT)
CALL PLOTVEL(EX,EY,1./BZI,TIME,NX,RY,1)
   IF (ICTRAN.GE.1000) THEN
      CALL PLOTTING(NE,1,NG1,POT,TIME,2)
   ENDIF
CALL CIMAGE(NE,NG1)
write(6,555) time,endtime,it

  c  end of plotting
  c
  do 140 j=1,ny
    do 140 i=1,nx
      ndens=ndens+ne(i,j)
    140 continue
    totenerg(iic)=ndens
  endif

  c  END TIME LOOP
  c

  1 continue

  c  plot field energy.
  c
  call plthst(totenerg,iic,1,'TOTAL ENERGY$').

  return
  end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

SUBROUTINE potential(nek,neq,uxbz,
  . exk,exq,eyk,eyq,phik,phiq,rk,rq,nx,ny,nz,it)

c calculates the potential phi by iterations. The iteration converges when
c the error criterion errmar is satisfied. Use phi to evaluate the rhs
c of the continuity equation

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

common/filter/ smk(64,128),smq(128)
common/efluid/ bzi,ifts
common/error:/ errmax

real nes(128,128), neinv(128,128)
real smk,smq
complex nek(64,128), neq(128), nesk(64,128), nesq(128)
complex exk(64,128), exq(128), eyk(64,128), eyq(128)
complex rk(64,128), rq(128), nekf(64,128), neqf(128)
complex phik(64,128),phiq(128)
complex neinvk(64,128),neinvq(128)
complex dnxk(64,128),dnxq(128),dnyk(64,128),dnyq(128)
complex txk(64,128),txq(128),tyk(64,128),tyq(128)
complex dphik(64,128),dphtxq(128),dphiyk(64,128),dphiyq(128)
complex sumk(94,128),sumq(128),sumkn(64,128),sumqn(128)
complex prphik(64,128), prphiq(128)
complex d2nk(64,128), d2nq(128)
complex phikmax(64,128)
parameter (tiny=1.e-30)
equivalence (nes,nesk)
equivalence (neinv,neinvk)

nx2=nx/2
ny2=ny/2

scale = nx*ny*nz
scale = 2./scale
scale2 = scale*scale

maxit = 100
kmax2=32**2

c assign initial phik to 0
c
if(ifts.eq.0) then
  do 5 j=1,ny
    phiq(j) = (0.,0.)
  do 5 i=1,nx2
  5    phik(i,j) = (0.0,0.0)
endif
c

c calculate the FFT of 1/nes

do 10 j=1,ny
  nesq(j) = neq(j)
  do 10 i=1,nx2
  10    nesk(i,j) = nek(i,j)
call rlft3(nes,nesq,nx,ny,nz,-1)
do 20 j=1,ny
  do 20 i=1,nx
  20    neinv(i,j) = 1./nes(i,j)/scale
call rlft3(neint,neinvq,nx,ny,nz,1)
c
Filter the electron density for upcoming convolution
c
do 30 j=1,ny
  neqf(j)=neq(j)*smq(j)
  neinvq(j)=neinvq(j)*smq(j)
  do 30 i=1,nx2
  30    nekf(i,j)=nek(i,j)*smk(i,j)
    neinvk(i,j)=neinvk(i,j)*smk(i,j)
continue
call derivk(nekf,neqf,dnxf,dnqf,nx,ny,nz,1)
call derivk(nekf, neqf, dnyk, dnyq, nx, ny, nz, 2)

C Begin iterations
C
do 200 it = 1, maxit

call derivk(phik, phiq, dpixk, dphixq, nx, ny, nz, 1)
call derivk(phik, phiq, dphik, dphiyq, nx, ny, nz, 2)

call convolve(dnxk, dnxq, dpixk, dphixq, scale2, txk, txq, nx, ny, nz)
call convolve(dnyk, dnyq, dphik, dphiyq, scale2, tyk, tyq, nx, ny, nz)

do 40 j = 1, ny
  sumq(j) = txq(j) + tyq(j) + uxbz*dnyq(j)
do 40 i = 1, nx2
  sumh(i, j) = txk(i, j) + tyk(i, j) + uxbz*dnyk(i, j)
  phikmax(i, j) = (tiny, tiny)
40 continue

call convolve(neinvk, neinvq, sumh, sumq, scale2,
               sumkn, sumqn, nx, ny, nz)
call laplacek(sumkn, sumqn, prphik, prphiq, nx, ny, nz, 1)

C Calculating error for convergence
C error criterion : l(phi max - phi)/phi maxl < errmax
C
do 50 j = 3, ny - 1
  do 50 i = 3, nx2
    if(phik(i, j).ne.(0., 0.)) then
      if((i**2+j**2).lt.kmax2 .or. (i**2+(ny-j)**2).lt.kmax2) then
        if (cabs(phik(i, j)).ge.cabs(phikmax(i1, j1))) then
          i1 = i
          j1 = j
          phikmax(i1, j1) = phik(i, j)
        endif
      endif
    endif
  enddo
50 continue
err = cabs((phikmax(i1, j1) - prphiq(i1, j1))/phikmax(i, j))

do 60 j = 1, ny
  phiq(j) = prphiq(j)
do 60 i = 1, nx/2
  phik(i, j) = prphiq(i, j)
60 continue

if(err.lt.errmax .and. ifts.ne.0) go to 100
ifts = 1
200 continue
100 call laplacek(nek, neq, d2nk, d2nq, nx, ny, nz, 2)

C
get Ex and Ey

call derivk(phik,phiq,exq,exq,nx,ny,nz,1)
call derivk(phik,phiq,eyk,eyq,nx,ny,nz,2)

do 110 j=1,ny
   exq(j)=-exq(j)
   eyq(j)=-eyq(j)
do 110 i=1,ny2
   exk(i,j)=-exk(i,j)
   eyk(i,j)=-eyk(i,j)
110 continue

c Calculate the right hand side of continuity equation (rk,rq)
c
call continuk
   .(nekf,neqf,exk,eyk,exq,eyq,d2nk,d2nq,rk,rq,nx,ny,nz)
   return
end

SUBROUTINE continuk
   .(nk,nq,exk,eyk,exq,eyq,d2nk,d2nq,rk,rq,nx,ny,nz)
c
Calculates the right hand side of the continuity eqn :
c
rk = - vxk*knek - vyk*kynek - Dk2nek
c where D = dc = diffusion constant
c
common /efluid/ bzi,ifts

complex nk(64,128), nq(128)
complex nxk(64,128),nxq(128),nyk(64,128),nyq(128)
complex exk(64,128),exq(128),eyk(64,128),eyq(128)
complex vxnxk(64,128),vxnxq(128),vynyk(64,128),vynq(128)
complex d2nk(64,128),d2nq(128)
complex rk(64,128),rq(128)

scale = nx*ny*nz
scale =2./scale
scale2=scale*scale*bzi
dc=0.15
dc=0.1

nx2=nx/2
ny2=ny/2

c Calculate derivatives in Fourier space of nx and ny
c
call derivk(nk,nq,nxk,nx,q,nx,ny,nz,1)
call derivk(nk,nq,nyk,nyq,nx,ny,nz,2)

C Calculate the convolution: ( = vxx*nxx )

C call convolve(eyk,eyp,nxk,nx, scale2,vxnxk,vxnxq,nx,ny,nz)
C Calculate the convolution: ( = vyk*nyk )
C call convolve(egk,egk,nyk,nyq,-scale2,vynyk,vynyk,nx,ny,nz)

C CALCULATE RHS OF CONTINUITY EQUATION :
C
C rk = -(vxx*nxx + vyk*nyk + dc*d2nk)
C
C do 40 j=1,ny
C rq(j) = -(vxxnq(j) + vynq(j) + dc*d2nq(j))
do 40 i=1,nx2
40 rk(i,j) = -(vxxn(i,j) + vynk(i,j) + dc*d2nk(i,j))

return
end
C
Subroutine fldinit(ng,1)
C calculates the Fourier, grid shifting and filter coefficients.
C Grid shifting and filter coefficients are taken from Pettersson and
C Orszag, 1971
C
C complex sppk,sppq,expc,imag,ip,jp,jm,ipc
real ksqi2,ksqiix,ksqiyl,ksqiy(64),ksqi1x(64)
real kks,kx,ly,1,ly1
real smk,smq
real kks,ykp,yym,kcrit,kmax,kmax2

ifs = 0
pi=4.*atan(1.)

nx2=ng/2
ny2=ng/2
lx=1
ly=1
dx=lx/ng
dy=ly/ng
dx2=dx/2
dy2=dy/2

kx=pi*2/lx
ky=pi*2/lx

C Calculate Fourier coefficients
C
do 60 j=1,ny2
   do 60 i=1,nx2
      ksq12(i,j)=.25/
      .
       (((sin(kx*dx2*i)/dx)**2+(sin(ky*dy2*j)/dy)**2)
   60      continue

   do 65 j=1,ny2
      ksqjy(j)=.25/((sin(ky*dy2*j)/dy)**2)
      ky(j)=sin(ky*dy*j)/dy
   65      continue

   do 70 i=1,nx2
      ksqix(i)=.25/((sin(kx*dx2*i)/dx)**2)
      kx(i)=sin(kx*dx*i)/dx
   70      continue

C Calculate grid shifting coefficients
C
imag = (0.0, -1.0)
px=pi/ng
py=pi/ng
ipc=ng/2
expc=cexp(img*px*ipc)

C
do 5 j=1,ng/2
   jp=j-1
   jm=-j
   sppq(j) =expc*cexp(img*py*jp)
   sppq(n+1-j)=expc*cexp(img*py*jm)
   do 5 i=1,ng/2
      ip=i-1
      sppk(i,j) =cexp(img*px*ip)*cexp(img*py*jp)
     sppk(i,ng+1-j)=cexp(img*px*ip)*cexp(img*py*jm)
   5      continue

C Calculate filter coefficients
C
rn=5
kmax=58
kmax2=kmax*kmax
kcri=ng/2

do 6 j=1,ng/2
  kyp=j-1
  kym=j
  smq(j) =exp(-((kcri**2+kyp**2)/kmax2)**rn)
  smq(ng+1-j)=exp(-((kcri**2+kym**2)/kmax2)**rn)
  do 6 i=1,ng/2
    kxp=i-1
    snk(i,j) =exp(-((kxp**2+kyp**2)/kmax2)**rn)
    snk(i,ng+1-j)=exp(-((kxp**2+kym**2)/kmax2)**rn)
  6 continue

return
end

ccc SUBROUTINE convolve(ek,eq,nk,nq,scale,vnk,vnq,nx,ny,nz)

ccc Calculates the convolution of two complex arrays by using the
ccc FFT convolution theorem and grid shift de-aliasing technique of
ccc
ccc real eshfs(128,128), nshfs(128,128)
ccc complex ek(64,128),nk(64,128),eq(128),nq(128)
ccc complex vnk(64,128),vnq(128)
ccc complex eshf(64,128),nshf(64,128),eshf(128),nshf(128)
ccc complex sppk,sppq
ccc common /dealias/ sppk(64,128),sppq(128)
ccc equivalence(eshfs,eshf)
ccc equivalence(nshfs,nshf)
ccc
ccc nx2=nx/2

do 20 j=1,ny
  vnq(j)=(0.0,0.0)
  do 20 i=1,nx2
    vnk(i,j)=(0.0,0.0)
  20 continue

ccc Calculate grid shifted quantities

  do 10 ishf = 1, 2
    if(ishf.eq.1) then
      do 1 j=1,ny

eshfq(j)=eq(j)*sppq(j)
nshfq(j)=eq(j)*sppq(j)
do 1 i=1,nx2
    eshfk(i,j)=ek(i,j)*sppk(i,j)
nshfk(i,j)=nk(i,j)*sppk(i,j)
1 continue
endif
if(ishf.eq.2) then
  do 2 j=1,ny
    eshfq(j)=eq(j)
nshfq(j)=eq(j)
do 2 i=1,nx2
    eshfk(i,j)=ek(i,j)
nshfk(i,j)=nk(i,j)
2 continue
endif

! Calculate convolution with the convolution theorem
! call rlf3(eshs,eshfq,nx,ny,nz,-1)
call rlf3(nshfs,nshfq,nx,ny,nz,-1)
do 5 j=1,ny
  do 5 i=1,nx
    eshsf(i,j)=eshfs(i,j)*nshfs(i,j)*scale
5 continue
call rlf3(eshs,eshfq,nx,ny,nz,1)

! Sum up convolutions of all shifted grid convolutions
if(ishf.eq.1) then
  do 6 j=1,ny
    vnq(j) = vnq(j) + .5*eshfq(j)*conjg(sppq(j))
do 6 i=1,nx2
    vnk(i,j) = vnk(i,j) + .5*eshfk(i,j)*conjg(sppk(i,j))
6 continue
endif
if(ishf.eq.2) then
  do 7 j=1,ny
    vnq(j) = vnq(j) + .5*eshfq(j)
do 7 i=1,nx2
    vnk(i,j) = vnk(i,j) + .5*eshfk(i,j)
7 continue
endif
10 continue
SUBROUTINE deriv(fk,fq,dfk,dfq,nx,ny,nz,ixy)

Returns the Fourier transformed $x$-derivative ($ikx*fk$) of the grid quantity if $ixy=1$ or the Fourier transformed $y$-derivative ($iky*fk$) of the grid quantity if $ixy=2$.

COMMON /fourcoeffd/ k1x(64), k1y(64)

COMPLEX fk(64,128), fq(128), kappa(64)
COMPLEX dfk(64,128), dfq(128)
REAL k1x, k1y

nx2=nx/2
ny2=ny/2

X-derivative

IF(ixy.EQ.1) THEN

DO 10 i=1,nx2
   kappa(i)=k1x(i)*(0.,-1.)

10 CONTINUE

DO 20 j=2,ny2
   dfq(j)=fq(j)*kappa(nx2)
   dfq(j+ny2)=fq(j+ny2)*kappa(nx2)
   dfk(1,j)=(0.,0.)
   dfk(1,j+ny2)=(0.,0.)
   dfk(j,ny2+1)=fk(j,ny2+1)*kappa(j-1)
   dfk(j,1)=fk(j,1)*kappa(j-1)

20 CONTINUE

dfk(1,1)=(0.,0.)
dfk(1,ny2+1)=(0.,0.)
dfq(1)=fq(1)*kappa(nx2)
dfq(ny2+1)=fq(ny2+1)*kappa(nx2)
ENDIF

Y-derivative

IF(ixy.EQ.2) THEN

DO 30 j=1,ny2
   kappa(j)=k1y(j)*(0.,-1.)

30 CONTINUE
continue
do 40 j=2,ny2
   dfq(j)=fq(j)*kappa(j-1)
   dfq(j+ny2)=-fq(j+ny2)*kappa(ny2+1-j)
   dfk(1,j)=fk(1,j)*kappa(j-1)
   dfk(1,j+ny2)=-fk(1,j+ny2)*kappa(ny2+1-j)
   dfk(j,ny2+1)=fk(j,ny2+1)*kappa(ny2)
   dfk(j,1)=(0.,0.,0.)
   do 40 i=2,nx2
      dfk(i,j)=fk(i,j)*kappa(j-1)
      dfk(i,j+ny2)=-fk(i,j+ny2)*kappa(ny2+1-j)
   40 continue
   dfk(1,1)=(0.,0.,0.)
   dfk(1,ny2+1)=fk(1,ny2+1)*kappa(ny2)
   dfq(1)=(0.,0.,0.)
   dfq(ny2+1)=fq(ny2+1)*kappa(ny2)
endif
return
end

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
SUBROUTINE laplacek(fk,fq,d2fk,d2fq,nx,ny,nz,idivide)

cc
Returns the Fourier transformed inverse Laplacian (fk/k) of the grid

cc quantity if idivide = 1 or the Laplacian (k fk) if idivide = 2
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
common /fourco2d/ ksqi2(64,64),ksqi1x(64),ksqi1y(64)

complex fk(64,128), fq(128)
complex d2fk(64,128), d2fq(128)
real ksqi2,ksqi1x,ksqi1y

nx2=nx/2
ny2=ny/2

if(idivide.eq.1) then
   do 1 j=2,ny2
      do 1 i=2,nx2
         d2fk(i,j)=fk(i,j)*ksqi2(i-1,j-1)
         d2fk(i,j+ny2)=fk(i,j+ny2)*ksqi2(i-1,ny2+1-j)
      1 continue
   do 2 i=1,nx2
      d2fk(i,ny2+1)=fk(i,ny2+1)*ksqi2(i-1,ny2)
      d2fk(i,1)=fk(i,1)*ksqi1x(i-1)
   2 continue
   d2fk(1,1)=(0.,0.,0.)
   d2fk(1,ny2+1)=fk(1,ny2+1)*ksqi1y(ny2)
end subroutine laplacek
do 3 j=2,ny2
   d2fk(1,j )=fk(1,j )*ksq1y(j-1)
   d2fk(1,j+ny2)=fk(1,j+ny2)*ksq1y(ny2+1-j)
   d2fq(j )=fq(j )*ksq12(nx2, j-1)
   d2fq(j+ny2)=fq(j+ny2)*ksq12(nx2,ny2+1-j)
3 continue

   d2fq( 1)=fq( 1)*ksq1x(nx2)
   d2fq(ny2+1)=fq(ny2+1)*ksq12(nx2,ny2)
else
   do 4 j=2,ny2
      do 4 i=2,nx2
         d2fk(i,j )=fk(i,j )/ksq12(i-1, j-1)
         d2fk(i,j+ny2)=fk(i,j+ny2)/ksq12(i-1,ny2+1-j)
4 continue
   do 5 i=1,nx2
      d2fk(i,ny2+1)=fk(i,ny2+1)/ksq12(i-1,ny2)
      d2fk(i, 1)=fk(i, 1)/ksq11x(i-1)
5 continue
   d2fk(1, 1)=(0.,0.)
   d2fk(1,ny2+1)=fk(1,ny2+1)/ksq1y(ny2)
   do 6 j=2,ny2
      d2fk(1,j )=fk(1,j )/ksq1y(j-1)
      d2fk(1,j+ny2)=fk(1,j+ny2)/ksq1y(ny2+1-j)
      d2fq(j )=fq(j )/ksq12(nx2, j-1)
      d2fq(j+ny2)=fq(j+ny2)/ksq12(nx2,ny2+1-j)
6 continue

   d2fq( 1)=fq( 1)/ksq1x(nx2)
   d2fq(ny2+1)=fq(ny2+1)/ksq12(nx2,ny2)
endif

return
end

SUBROUTINE plotting (ne,l,ng1,title,time,iplot)
  c
  c returns the contour plot of iplot=1, or the surface plot otherwise
  c
  character*32 title
  character*7 tim
  real work(190,130,3)
  tcor=time
  if (iplot.eq.1) then
call prtplt(ne,flo,fhi,finc)
call plotf2d(ne,flo,fhi,finc,title,tcor,1,ng1)
else
  write(tim,100) tcor
endif

format(f7.3)
call set(0.0, 1.0, 0.0, 1.0, 1.0, 1024.0, 1.0, 1024.0, 1)
call pwrizx(512.0,964.0,title,32,1,0,0)
call pwrizx(512.0,900.0,'T = ',4.0,0,1)
call pwrizx(512.0,900.0,tim ,7.0,0,0,1)
call ezsrfc(ne,ng1,ng1,-55..35.,work)
call *zsrfc(ne,ng1,ng1,135..15.,work)
return
end

cccccccccccccccccccccccccccccccccccccccccccccccccc
subroutine plotvel(ex,ey,bz,time,nx,ny,nx)
cccccccccccccccccccccccccccccccccccccccccccccccccc
real ex(129,129), ey(129,129)
real vxs(32,32), vys(32,32)
character*7 tim
tcor=time

write(tim,100) tcor

format(f7.3)
do 10 j=1,ny
do 10 i=1,nx
  if(mod(i,4).eq.0 .and. mod(j,4).eq.0) then
    ii=i/4
    jj=j/4
    vxs(ii,jj)=ey(i,j)*bz
    vys(ii,jj)=-ex(i,j)*bz
  endif

10 continue

call set(0.0, 1.0, 0.0, 1.0, 1.0, 1024.0, 1.0, 1024.0, 1)
call pwrizx(512.0,1000.0,'VELOCITY FIELD',14,1,0,0)
call pwrizx(512.0,940.0,'T = ',4.0,0,1)
call pwrizx(512.0,940.0,tim ,7.0,0,0,-1)
call velvct(vxs,32,vys,32,32,32,flo,fhi,0,0,0,0)
call frame

return
end

cccccccccccccccccccccccccccccccccccccccccccccccccc
subroutine plotf2d(f,flo,fhi,finc,title,time,1,ng1)
cccccccccccccccccccccccccccccccccccccccccccccccccc
plots the x-y coordinate space of fields at specified times
real f(129,129)
real rwrk(20000)
real 1
integer iwrk(8000)
character*7 tim
character*32 title

write(tim,100) time
100 format(f7.3)

NCAR GRAPHICS CALLS

call sst(0.0, 1.0, 0.0, 1.0,1.0, 1024.0, 1.0, 1024.0,1)
call pwritx(512.0,964.0,title ,32,1,0, 0)
call pwritx(512.0,900.0,'T = ', 4,0, 0, 1)
call pwritx(512.0,900.0,tim , 7,0,0,-1)
call pwritx(512.0,64.0,'X',1.0, 0, 0)
call pwritx(48.0,512.0,'Y',1.0,90,0)

c = (fhi-flo)/finc + 1
call cpseti('CLS - CONTOUR LEVEL SELECTION',0)
call cpseti('NCL - NUMBER OF CONTOURING LEVELS',nc)

do 1 ic = 1,nc
call cpseti('PAI - PARAMETER ARRAY INDEX',ic)
value = (ic-1)*finc + flo
call cpsetr('CLV - CONTOUR LEVEL VALUE',value)
call cpseti('CLU - CONTOUR LEVEL USE',1)
if(value.lt.0.0).
call cpseti('CLD - CONTOUR LINE DASH PATTERN',21845)
if(value.ge.0.0).
call cpseti('CLD - CONTOUR LINE DASH PATTERN',65535)
1 continue

lrwrk = 20000
liwrk = 8000
call cpsetr('XC1 - X COORDINATE AT INDEX 1',0.)
call cpsetr('XCM - X COORDINATE AT INDEX #', 1)
call cpsetr('YC1 - Y COORDINATE AT INDEX 1',0.)
call cpsetr('YCM - Y COORDINATE AT INDEX #', 1)
call set(.15,.85,.15,.85,0.,1,0,.1,1)
call cpseti('SET - DO SET CALL FLAG',0)
call cprect(f,ng1,ng1,ng1,lrwrk,liwrk,iwrk)
call labmod('f4.0','f4.0','f4.0',1,4,4,2,2,0,0,0)
call periml(4,4,4,4)
call cpcldr(f,lrwrk,iwrk)
call cpsetr('CLD - CONTOUR INTERVAL USED',finc)
call cpsetc('HLT - HIGH/LOW LABEL TEXT STRINGS',' ')
call cpseti('ILP - INFORMATIONAL LABEL POSITIONING FLAG',0)
call cpsetr('ILX - INFORMATIONAL LABEL X COORDINATE', .5)
call cpsetr('ILY - INFORMATIONAL LABEL Y COORDINATE',-.2)
call cplbdr(f,lrwrk,iwrk)
call frame
return
end

real f(129,129)
flo= f(1,1)
fin= flo
do 1 iy=1,129
do 1 ix=1,129
fin= amax1( f(ix,iy),fin )
1 flo= amin1( f(ix,iy),flo )
fin=(fin-flo)/14.
return
end

SUBROUTINE fourn(data,nn,ndim,isign)

Replaces data by its ndim-dimensional discrete Fourier transform, if
isign is input as 1. nn(1:ndim) is an integer array containing the
lengths of each dimension (number of complex values), which MUST be
powers of 2. data is a real array of length twice the product of
these lengths, in which the data are stored as in a multidimensional
complex FORTRAN array. If isign is input as -1, data is replaced by
its inverse transform times the product of the lengths of all
dimensions. This routine is taken from "Numerical Recipes" by Press
et al.

INTEGER isign,ndim,nn(ndim)
REAL data(*)
INTEGER i1,i2,i2rev,i3,i3rev,ibit,jdim,ip1,ip2,ip1,ip2,
     ,ip3,k1,k2,n,nprev,nrem,ntot
REAL tempi,tempr
DOUBLE PRECISION theta,wi,wpi,wpr,wr,wtemp

ntot=1
do 11 idim=1,ndim
   ntot=ntot*nn(idim)
11 enddo
nprev=1
do 18 idim=1,ndim
   n=nn(idim)
   nrem=ntot/(n*nprev)
   ip1=2*nprev
   ip2=ip1*n
   ip3=ip2*nrem

129
i2rev=1
do 14 i2=1,ip2,ip1
   if (i2.lt.i2rev) then
      do 13 i1=i2,i2+ip1-2,2
         do 12 i3=i1,ip3,ip2
            i3rev=i2rev+i3-i2
            temp=data(i3)
            tempi=data(i3+1)
            data(i3)=data(i2rev)
            data(i3+1)=data(i3rev+1)
            data(i3rev)=temp
            data(i3rev+1)=tempi
      enddo
   endif
   ibit=ip2/2
   if ((ibit.ge.ip1).and.(i2rev.gt.ibit)) then
      i2rev=i2rev-ibit
      ibit=ibit/2
   goto 1
   endif
14 enddo
if(ip1=ip1
2 if(ipf1.lt.ip2) then
   ipf2=2*ipf1
   theta=sign*6.28318530717959d0/(ipf2/ip1)
   wpr=-2.d0*sin(0.5d0*theta)**2
   wpi=sin(theta)
   wr=1.d0
   wi=0.d0
   do 17 i3=1,ipf1,ip1
      do 16 i1=i3,i3+ip1-2,2
         do 15 i2=i1,ip3,ipf2
            k1=i2
            k2=k1+ipf1
            temp=sngl(wr)*data(k2)-sngl(wi)*data(k2+1)
            tempi=sngl(wr)*data(k2+1)+sngl(wi)*data(k2)
            data(k2)=data(k1)-temp
            data(k2+1)=data(k1+1)-tempi
            data(k1)=data(k1)+temp
            data(k1+1)=data(k1+1)+tempi
15 enddo
16 enddo
   wtemp=wr
   wr=wr*wpr-wi*wpi+wr
   wi=wi*wpr+wtemp*wpi+wi
17 enddo
if(ip1=ipf2
2 goto 2
endif
nprev=n*nprev
18 enddo
return
END
SUBROUTINE rlft3(data,speq,n1,n2,n3,isign)

c Given a two- or three-dimensional real array data whose dimensions
are n1,n2,n3 (where n3 is 1 for the case of a two-dimensional
array), this routine returns (for isign=1) the complex fast
Fourier transform as two complex arrays: On output, data contains
the zero and positive frequency values of the first frequency
component, while speq contains the Nyquist critical frequency values
of the first frequency component. Second (and third) frequency
components are stored for zero, positive, and negative frequencies,
in standard wrap-around order. For isign=-1, the inverse transform
(times n1*n2*n3/2 as a constant multiplicative factor) is
performed, with output data (viewed as a real array) deriving from
input data (viewed as complex) and speq. The dimensions n1,n2,
and n3 must always be integer powers of 2. This routine is taken from
"Numerical Recipes" by Press et al.

INTEGER isign,n1,n2,n3
COMPLEX data(nn1/2,nn2,nn3),speq(nn2,nn3)
INTEGER i1,i2,i3,j1,j2,j3,nn(3)
DOUBLE PRECISION theta,wi,wpi,wpr,wr,wt,wtm
COMPLEX c1,c2,h1,h2,w

c1=cmplx(0.5,0.0)
c2=cmplx(0.0,-0.5*isign)
theta=6.283185307179589d0/4*isign+n1
wpr=-2.0d0*sin(0.5d0*theta)**2
wpi=sin(theta)
nn(1)=n1/2
nn(2)=n2
nn(3)=n3
if (isign.eq.1) then
  call fourn(data,nn,3,isign)
do 13 i3=1,nn3
     do 12 i2=1,nn2
        speq(i2,i3)=data(1,i2,i3)
 11     enddo
 12   enddo
endif
do 15 i3=1,nn3
  j3=1
  if (i3.ne.1) j3=nn3-i3+2
  wr=1.0d0
  wi=0.0d0
do 14 i1=1,nn1/4+1
     j1=nn1/2-i1+2
 13     do 12 i2=1,nn2
        j2=1
        if (i2.ne.1) j2=nn2-i2+2
        if (i1.eq.1) then
          h1=c1*(data(1,i2,i3)+conjg(speq(j2,j3)))+
h2=c2*(data(1,i2,i3)-conjg(speq(j2,j3)))
data(1,i2,i3)=h1+h2
enddo
speq(j2,j3)=conjg(h1-h2)
else
  h1=c1*(data(i1,i2,i3)+conjg(data(j1,j2,j3)))
  h2=c2*(data(i1,i2,i3)-conjg(data(j1,j2,j3)))
  data(i1,i2,13)=h1*w*h2
  data(j1,j2,j3)=conjg(h1-w*h2)
endif
enddo
wend
wtemp=wr
wr=wr*wpr-wi*wpi+wr
wi=wi*wpr+wtemp*wpi+wi
w=cmplx(sngl(wr),sngl(wi))
enddo
endif
if (isign.eq.-1) then
  call fourn(data,nn,3,isign)
endif
return
END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

function ran1(idum)

  function RAN1 computes a uniform random number between 0 and 1.
  algorithm described in "Numerical Recipes" by Press et al.
  Set IDUM to any negative value to initialize.
  
real r(97)
parameter (m1=259200,ia1=7141,ic1=54773,rm1=1./m1)
parameter (m2=154456,ia2=8121,ic2=28411,rm2=1./m2)
parameter (m3=243000,ia3=4561,ic3=51349)
data iff /0/
if(idum.lt.0.or.iff.eq.0) then
  iff=1
  ix1=mod(ic1-idum,m1)
  ix1=mod(ia1*ix1+ic1,m1)
  ix2=mod(ix1,m2)
  ix1=mod(ia1*ix1+ic1,m1)
  ix3=mod(ix1,m3)
doi j=1,97
  ix1=mod(ia1*ix1+ic1,m1)
  ix2=mod(ia2*ix2+ic2,m2)
  r(j)=(float(ix1)+float(ix2)*rm2)*rm1
continue
idum=1
endif
ix1=mod(ia1*ix1+ic1,m1)
ix2=mod(ia2*ix2+ic2,m2)
ix3=mod(ia3*ix3+ic3,m3)
j=1+(97*ix3)/m3
if(j.gt.97.or.j.lt.1)pause
ran1=r(j)
r(j)=(float(ix1)+float(ix2)*rm2)*rm1
return

132
end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
subroutine cimage(f,ng1)
c
c write to aa image file for animation
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
real f(129,129), fs(129,129)
integer image(129,129)
c
c get maximum and minimum values to scale f(x,y)
c
  fmin=f(1,1)
  fmax=fmin
do 1 iy=1,ng1
do 1 ix=1,ng1
    fmax=max1(f(ix,iy),fmax)
    fmin=min1(f(ix,iy),fmin)
1 continue

scale=255./(fmax-fmin)
do 2 iy=1,ng1
do 2 ix=1,ng1
  fs(ix,iy)=f(ix,iy)
  if (fs(ix,iy).gt.fmax) fs(ix,iy)=fmax
  if (fs(ix,iy).lt.fmin) fs(ix,iy)=fmin
2 continue
c
cc scale f to integers from 0 to 255
c
do 3 iy=1,ng1
do 3 ix=1,ng1
  image(ix,iy)=(fs(ix,iy)-fmin)*scale
3 continue
c
c write out image file
c
10 format(129(1x,i3))
write(20,10) ((image(ix,iy), ix=1,ng1), iy=1,ng1)
return
end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
subroutine pltshst(energ,nitem,many,label)
c
c plot time history
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
real energ(nitem)
character*30 label
NCAR graphics calls

call agsetc('LABEL/NAME.', 'L')
call agseti('LINE/NUMBER.', 80)
call agsetc('LINE/TEXT.', '$')
call agsetc('LABEL/NAME.', 'B')
call agseti('LINE/NUMBER.', -80)
call agsetc('LINE/TEXT.', 'Time$')
call ezmy(energ,nitem,many,nitem,label)

return
end

c


Appendix B

MOVIE program listing

MOVIE.F is used to convert a DENSITY IMAGE to a DENSITY.HDF file for density animation. The animation is generated with the NCSA software XDataSlice. If the number of frames to be animated is changed in the subroutine CIMAGE in ESSPEC2, be sure to change the dimension of variable ivals and image below. The size for HDF files are usually large (> 4 MB) so that it is recommended to store only the IMAGE files.

The listing of the program MOVIE is as follows

```fortran
*****%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*
*  program movis.f
*
*  converts the DENSITY IMAGE to DENSITY.HDF for density animation using
  *XDataSlice (XDS).
*
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

integer ivals(129,129,63)
real image(129,129,63)
integer shape(3),ret
integer DFSDsetdims
integer DFSDputdata

open(5, file='density.image', status='old', err=888)
shape(1)=129
shape(2)=129
shape(3)=63
c shape(3) is the number of frames to be animated

c read in integer values from input file
  c
  ix=shape(1)
iy=shape(2)
it=shape(3)

do 100 k=1,ix
    read(5,10)((ivals(i,j,k),i=1,ix),j=1,iy)
  100 continue

10 format(129(ix,i3))
do 200 k=1,ix
    do 200 j=1,iy
        do 200 i=1,ix
            image(i,j,k)=ivals(i,j,k)

200 continue
```

135
c
write image to an HDF file

c
ret=DFSDsetdims(3,shape)
ret=DFSDputdata('density.hdf',3,shape,image)
if (ret.ne.0) then
  write(*,'(a)')'Error Writing HDF File'
endif

888  stop
end
References


Huba, J. D., S. L. Ossakow, P. Satyanarayana, and P. N. Guzdar, Linear theory of the $\mathbf{E} \times \mathbf{B}$ instability with an inhomogeneous electric field, J. Geophys. Res., 88, 425, 1983.


Mendillo, Michael, Ionospheric holes: a review of theory and recent experiments, COSPAR, Active experiments, 8, 51, 1988.


Vita

Atikah Ismail was born in Jakarta, Indonesia on October 7, 1968. She came to the United States in 1984 and graduated from Richmond Community High School in Richmond, Virginia in 1987. She attended Virginia Tech during which she was involved with the Fiber and Electro-Optic Research Center. She received her B.S. in Electrical Engineering in 1991 and decided to pursue a Master degree. She joined the electromagnetic research group where, in 1992, she had the privilege to serve under Dr. Wayne Scales as a graduate research assistant working on numerical simulation of ionospheric processes (specifically the $\mathbf{E} \times \mathbf{B}$ interchange instabilities) using Fourier spectral methods. In summer 1993 she had the opportunity to intern at AT & T Bell Labs in Holmdel, New Jersey, working with erbium-doped fiber amplifiers. Her areas of interest include electromagnetics, plasma simulation, fiber optics and communications.

[Signature]

[Atikah Ismail]