ECE 6931 Master of Engineering Design

Applying Automatic Differentiation to Ray Tracing Problems in Radio

MEng Design Project Report

Author: Shaotian Zhang

StudentID: sz442

Project Advisor: David Hysell

(Spring, 2023)

Cornell University, Ithaca

May 19, 2023

Contents

1 Abstract

This study addresses the need for enhanced accuracy in modeling radio wave propagation through the ionosphere, a medium rich in free electrons that significantly impact radio signals. Traditional approaches, utilizing direct variational sensitivity analysis, often entail complex derivative computations. These complexities are intensified when considering realistic dispersion relations or the usage of curvilinear coordinates.

In an attempt to circumvent these computational challenges, the proposed approach integrates automatic differentiation (AD) with a stiff ordinary differential equation solver. The aim is to optimize the prediction of radio ray trajectories based on the control parameter, specifically the initial radio ray direction. However, the integration of AD within a gradient descent method for refining the ionospheric model has presented certain challenges that have yet to be fully surmounted.

Despite these challenges, the investigation offers a potential pathway to improved computational efficiency and accuracy in ionospheric radio wave propagation modeling. The Python code provided as part of this study serves as an initial step towards demonstrating the practical application of these concepts. Future work will focus on further refining this methodology and addressing the current predicaments.

2 Introduction

The Earth's atmospheric structure, particularly the ionosphere, plays a pivotal role in the intricate dynamics of wave propagation. This ionized layer, extending from the upper Mesosphere to the Exosphere, contains a dense concentration of ions and free electrons that significantly impact the behavior of electromagnetic waves, such as radio waves. Comprehending these interactions is vital for areas like geophysics and atmospheric sciences, and also has profound implications for telecommunications and space exploration.

This report aims to illuminate the mathematical and computational complexities associated with analyzing wave propagation in the ionosphere. Specifically, it emphasizes the role of Python as a robust computational tool in this exploration, with a particular focus on Geometric (Hamiltonian) Optics.

Wave propagation in complex media such as the ionosphere can be represented using Partial Differential Equations (PDEs). Given the inherent heterogeneity of these media, solving these PDEs is not straightforward. A practical approach to mitigating this complexity is by converting these PDEs into a system of Ordinary Differential Equations (ODEs), thereby simplifying the wave problem into a more manageable particle problem.

Additionally, the report underscores the utilization of the variational method for sensitivity analysis, a technique that aids in predicting how the behavior of wave propagation responds to specific control parameters, thereby providing valuable insights into the system's dynamics.

For addressing the high degree of complexity and precision required in these problems, the concept of Automatic Differentiation (AD) is introduced. AD transcends the limitations of traditional differentiation methods by applying the chain rule to the basic arithmetic operations and elementary functions executed by computer programs, resulting in an accurate and efficient computation of derivatives, a fundamental aspect of understanding wave propagation.

An attempt was made to further refine the analysis by optimizing the ionospheric model using actual data (i.e., ray traveling time from transmitter to receiver) and simulation data derived from Python code. Despite the potential of this approach, the application of JAX's grad() function, a key component of Automatic Differentiation, encountered difficulties due to the discrete nature of the diffrax solver. However, these challenges serve as valuable lessons and pave the way for future explorations.

In summary, this report provides a comprehensive overview of the methodologies and computational tools central to understanding wave propagation in the ionosphere. Through a focus on Python, Geometric Optics, the variational method, and Automatic Differentiation, it aims to set a robust foundation for future scientific exploration and technological advancements in this field.

3 Ionospheric Fundamentals: Composition, Structure, and Role in Wave Propagation

The ionosphere is a key layer of the Earth's atmosphere that significantly influences the behavior and propagation of radio waves. Comprising parts of the Mesosphere, Thermosphere, and Exosphere, the ionosphere is distinctive due to its high concentration of free electrons and ions. The ionization process, largely driven by solar radiation interacting with atmospheric gases, accounts for this unique composition.

The structure of the ionosphere is dynamic, with its composition and density varying according to altitude, time, geographical location, and solar activity. These variations result in the ionosphere being segmented into sub-layers, namely the D, E, and F layers, each having unique ionization and electron density profiles.

This ionized region of Earth's atmosphere plays a critical role in the propagation of radio waves. The free electrons within the ionosphere interact with these waves, causing phenomena like reflection, refraction, absorption, and scattering. Consequently, the state and behavior of the ionosphere can have profound effects on radio communications, navigation systems, and satellite operations.

Understanding the ionosphere's properties and its effects on radio wave propagation is essential. It is not only significant for scientific studies in geophysics and atmospheric sciences, but also finds extensive applications in telecommunications and space exploration. This knowledge forms the foundation for exploring wave propagation complexities within the ionosphere and for developing robust mathematical and computational tools to solve associated problems.

4 Geometric Optics: A Hamiltonian Perspective

The study of wave dynamics, fundamental to diverse fields of physics, is conventionally delineated by the formulation of partial differential equations (PDEs). These equations provide the mathematical underpinning to describe the behavior of waves in various media, each type of wave being governed by a specific set of equations.

In the realm of fluid dynamics, waves propagating in fluids are regulated by the fluid equations (refer to Figure 1). These equations encapsulate the balance of momentum, mass, and energy, thereby serving as the theoretical backbone for understanding wave propagation in fluid media.

Shifting the focus to electromagnetism, the behavior of electromagnetic waves is illustrated by Maxwell's equations (see Figure 2). These four integral-differential equations succinctly describe how electric and magnetic fields interact, thereby dictating the properties and propagation of electromagnetic waves.

Lastly, in the geophysical context, mechanical waves such as seismic waves traversing through the Earth are governed by the seismic wave equation (as demonstrated in Figure 3). This equation, a specific form of the wave equation, is essential for comprehending the propagation of seismic waves in the Earth's crust, enabling the prediction of wave behavior during seismic events.

By examining these different wave equations, we can develop a broader understanding of wave dynamics across various physical domains.

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\
&\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes (\rho \mathbf{u})) + \nabla p = 0 \\
&\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u}(E + p)) = 0,
\end{aligned}$$

Figure 1: Fluid Equations

$$\begin{aligned}
\nabla \cdot \mathbf{E} = 0, \qquad & \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \\
\nabla \cdot \mathbf{B} = 0, \qquad & \nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}.
\end{aligned}$$

Figure 2: Maxwell's Equations

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Figure 3: One-Dimensional Wave Equation

While the resolution of partial differential equations (PDEs) can be relatively straightforward when dealing with homogeneous media and linear systems, thanks to the availability of a known dispersion relation (D), the scenario becomes markedly more complex when addressing inhomogeneous media. Here, due to the spatially varying parameters, the eigenfunctions diverge from the simplicity of plane waves, thereby adding a layer of intricacy to the solution process.

In this work, a significant transformation has been implemented to overcome this complexity: the conversion of the original system of PDEs into a system of ordinary differential equations (ODEs). This reformulation not only simplifies the computational problem but also reorients the conceptual framework of the approach. Rather than treating waves within an inhomogeneous medium, the focus shifts towards treating particles, as depicted in Figure 4. This change of perspective is critical in simplifying the computation and interpretation of wave behavior in complex media.

$$
\begin{aligned}
\frac{d\mathbf{k}}{dt} &= -\frac{\partial D/\partial \mathbf{x}}{\partial D/\partial \omega} \\
\frac{d\mathbf{x}}{dt} &= \frac{\partial D/\partial \mathbf{k}}{\partial D/\partial \omega} \\
\frac{d\omega}{dt} &= -\frac{\partial D/\partial t}{\partial D/\partial \omega}
\end{aligned}
$$

Figure 4: System of ODEs

5 Variational (Forward) Method for Sensitivity Analysis

The variational (forward) method illuminates how control parameters, such as the azimuth or elevation of the initial wavevector, shape ray trajectories in the ionosphere. A system of variational sensitivity equations, derived from the original ray tracing equations, traces the evolution of state vector sensitivities to these control parameters.

On solving this system of Ordinary Differential Equations (ODEs) over a given time frame, the state variables and their sensitivities at a particular time can be obtained. The initial conditions for these sensitivity equations depend on the control parameter, and total derivatives with respect to these parameters are evaluated using the chain rule.

While this method is powerful, complexities arise when applying it beyond Cartesian coordinates. Additionally, calculating second derivatives of the dispersion relation can be non-trivial, often necessitating the use of symbolic computer mathematics. Despite these challenges, the variational (forward) method remains an essential tool for developing accurate models of ray propagation in the ionosphere.

$$\frac{d}{dq}\frac{d\mathbf{x}}{dt} = \frac{d}{dt}\frac{d\mathbf{x}}{dq}, \quad \frac{d}{dq}\frac{d\mathbf{k}}{dt} = \frac{d}{dt}\frac{d\mathbf{k}}{dq}$$

$$\begin{aligned}
\frac{d}{dt}\frac{d\mathbf{k}}{dq} &= -\frac{D_\omega D_{\mathbf{x},q} - D_{\mathbf{x}} D_{\omega,q}}{D_\omega^2} \\
\frac{d}{dt}\frac{d\mathbf{x}}{dq} &= \frac{D_\omega D_{\mathbf{k},q} - D_{\mathbf{k}} D_{\omega,q}}{D_\omega^2}
\end{aligned}$$

Figure 5: Sensitivity of Parameter q

In the Python script titled 'rays-numpy.py' (refer to Code Appendix), both the dispersion relation and electron density are known variables. Given these initial values, the script calculates all the derivatives manually before applying an ODE solver to them. This process results in the generation of the wave propagation line depicted in Figure 6.

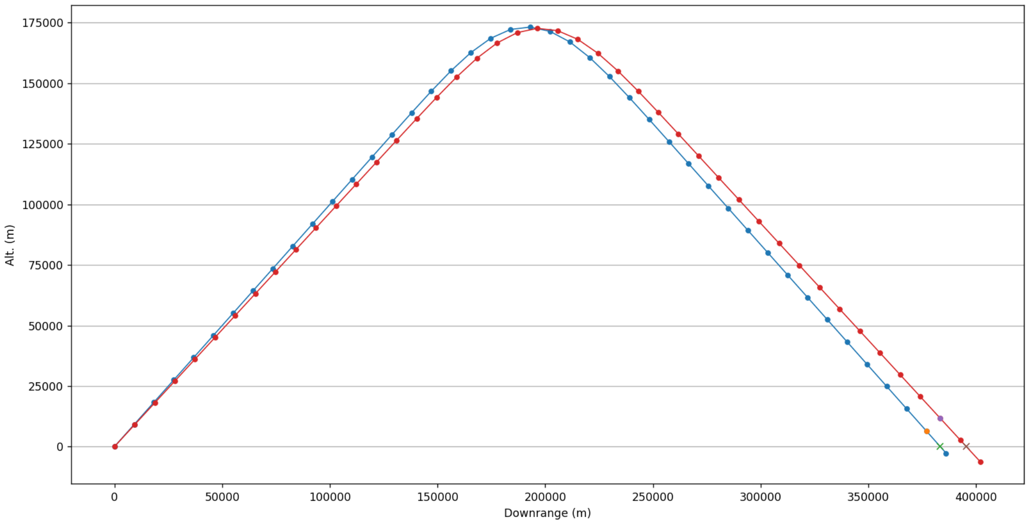

Figure 6: Wave Propagation

However, this approach can be error-prone and labor-intensive, as demonstrated even in the simplest case(Code shown below).

```
def prime(t, p):
    [x, z, kx, kz, xq, zq, kxq, kzq] = p
    nez = ne(z+0.5)-ne(z-0.5)  # finite difference
    nezz = (ne(z+1)-2*ne(z)+ne(z-1))
    hw = -(r*ne(z)+(kx**2+kz**2)*c**2)/w**3
    hx = 0.0
    hz = (0.5*r/w**2)*nez
    hkx = (c/w)**2*kx
    hkz = (c/w)**2*kz
    hw2 = hw*hw
    hwq = -zq*(2/w**3)*nez - kxq*(2*c**2/w**3)*kx - kzq*(2*c**2/w**3)*kz
    hkxq = kxq*(c/w)**2
    hkzq = kzq*(c/w)**2
```

```python
    hxq = 0
    hzq = zq*(r*0.5/w**2)*nezz
    pp = [-c**-1*hkx/hw, -c**-1*hkz/hw, c**-1*hx/hw, c**-1*hz/hw, -c**-1
   *
          (hw*hkxq-hkx*hwq)/hw2, -c**-1*(hw*hkzq-hkz*hwq)/hw2, c**-1*(hw*
   hxq-hx*hwq)/hw2, c**-1*(hw*hzq-hz*hwq)/hw2]
    return (pp)
```

Listing 1: Python Code

To address these challenges, an alternative approach from the field of automatic differentiation has been implemented.

6 Automatic Differentiation

Automatic Differentiation (AD) represents an ensemble of algorithmic methodologies, specifically developed to evaluate the derivative of a function embodied within a computational program. An intrinsic trait of any such program, regardless of its complexity, is the execution of a series of fundamental arithmetic operations, including addition, subtraction, multiplication, division, and the use of elementary functions such as exponential, logarithmic, sine, cosine functions.

AD leverages this elemental characteristic, systematically applying the chain rule of differentiation to the underlying operations. This allows AD to accommodate the computation of derivatives of any order, delivering results with high precision and efficiency. The process imposes a minimal computational burden, necessitating only a marginal increase in arithmetic operations compared to the original program.

As a consequence of its precision and automated nature, AD has emerged as a potent instrument in the realm of computational mathematics and scientific computing. Its accurate, efficient derivative computation offers substantial benefits in tackling complex mathematical problems, aiding in the continual advancement of these fields.

6.1 Utilization of JAX for Efficient Derivative Computation in Ionospheric Studies

JAX, standing for Just After eXecution, brings about a revolution in the domain of high-performance machine learning research by enabling execution of NumPy operations on various computational platforms including CPUs, GPUs, and TPUs. JAX's ability to marry the automatic differentiation capabilities of Autograd with the high-performance linear algebra computations of XLA provides a robust toolkit for a variety of scientific computing needs.

An instrumental feature of JAX's capabilities is its automatic differentiation function, 'grad', which eliminates the need for laborious manual calculations of derivatives. This function enables a highly efficient and accurate computational methodology, particularly vital for complex scientific computations and machine learning applications.

In the context of ionospheric studies, the 'grad' function in JAX has been utilized to streamline the computational process associated with derivative calculations. The transformation of complex partial differential equations (PDEs) into more tractable ordinary differential equations (ODEs) often necessitates derivative computations. Traditionally, these derivatives would be computed manually, a process that is not only time-consuming but also prone to human error.

Employing the 'grad' function in JAX to compute these derivatives automates this process, thereby ensuring accuracy and enhancing computational efficiency. This automated calculation process significantly reduces the possibility of errors introduced during manual computations and allows for more precise modeling and predictions in ionospheric studies.

The use of JAX, and more specifically its 'grad' function, represents a substantial advancement in the approach to derivative computations within the study of the ionosphere. This enhanced methodology provides a more reliable and efficient way of understanding the complex dynamics at play in the ionosphere, thereby enriching the overall knowledge base and paving the way for future explorations and discoveries in the field.

6.2 Employing Diffrax for Numerical Resolution of Transformed Differential Equations

The transformation of complex Partial Differential Equations (PDEs) into a more tractable system of Ordinary Differential Equations (ODEs) necessitates a robust method to find their solutions. Given the nonlinearity and complexity inherent in these systems, analytical solutions are often impractical or impossible to derive. Hence, numerical methods become indispensable, providing approximations to the solutions that can be improved iteratively to achieve a desired level of accuracy.

Diffrax, a Python-based computational library specializing in differential equation solving, plays a central role in this process. Its distinctive feature lies in its adeptness at addressing problems characterized by discrete data or discretized descriptions of the system under study.

In the context of the wave propagation problem in the ionosphere, the use of Diffrax becomes particularly relevant. After converting the initial PDE system - a mathematical representation of wave propagation - into a system of ODEs, Diffrax is employed to solve the transformed ODE system. The strength of Diffrax in this scenario stems from its ability to handle discretized versions of the system, allowing for the exploration of system dynamics in a step-by-step fashion.

This process is a numerical approximation approach, rather than an analytical solution method, thereby aligning perfectly with Diffrax's operational paradigm. By employing Diffrax for solving the system of ODEs, a discrete solution set is obtained. This set forms the basis for analyzing and understanding the phenomena of wave propagation in the ionosphere.

In essence, Diffrax serves as a valuable tool in unraveling the complexities of wave propagation in the ionosphere, enabling the study of system dynamics through the numerical resolution of transformed differential equations. This approach provides a practical and efficient pathway to understanding and predicting wave behavior in this critical atmospheric layer.

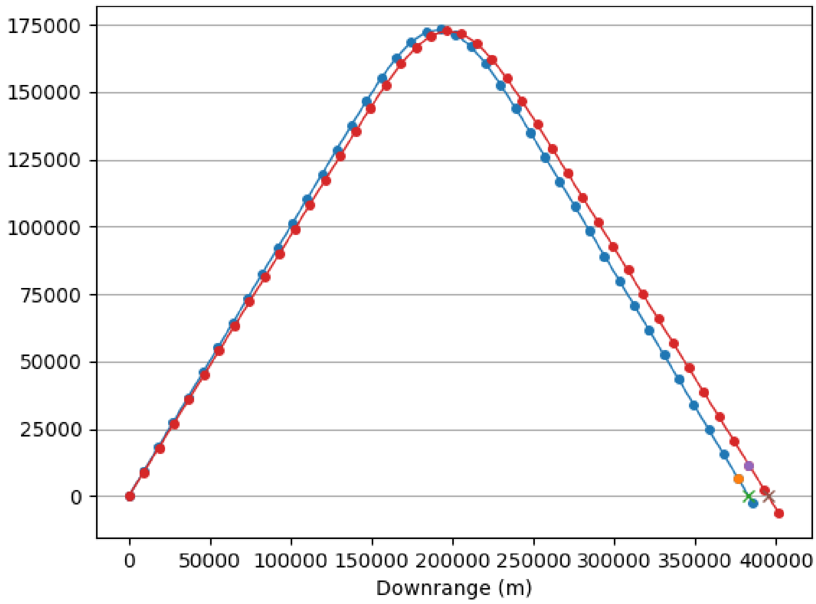

Figure 7: Results for Diffrax Solver

6.3 Implementing GPU-Accelerated Automatic Differentiation with JAX in Ionospheric Modelling

A pivotal step in this research was the application of JAX's 'grad' function in calculating derivatives, leveraging Google Colab's GPU acceleration. This computational environment offered an advanced infrastructure for handling the involved scientific computations, instrumental in achieving accurate and efficient results.

In the course of the research, the computational model of ionospheric wave propagation was fed into the 'grad' function. This process essentially translated the complex model into a set of derivatives, replacing the traditionally laborious manual calculations with an automated, more precise procedure. To harness the high-performance computation capabilities of Google Colab's GPUs, the execution of this process was configured to be GPU-accelerated.

This GPU-accelerated application of JAX for automatic differentiation significantly streamlined the computational workflow. Not only did it deliver high precision results, but it also paved the way for handling more sophisticated sensitivity analysis and optimization tasks, underlining its potential for future high-performance computing applications in this field.

While the execution time was observed to be longer than conventional NumPy computations, the improved accuracy and efficiency of the results rendered this trade-off acceptable. This aspect of the study highlights the potential for future research into optimizing computational performance without compromising the accuracy afforded by automatic differentiation.

7 Challenges and Breakthroughs in the Optimization of the Ionospheric Model

The culmination of the research efforts revolved around the optimization of coefficients within the ionospheric model. This stage involved synthesizing both actual data and simulation data to facilitate a comprehensive understanding of the system's behavior.

The research approach was methodical: a variety of rays, spanning angles from 45 to 55 degrees, were simulated. The resulting wave propagation data was then integrated to compute the corresponding path lengths. Following this, the total travelling time of each ray was calculated by dividing the path length by the speed of light 'c'. This step, though seemingly straightforward, underscored a significant challenge: the inhomogeneity of electron density within the ionosphere could lead to variations in the speed of light, a factor that could potentially impact the accuracy of the model.

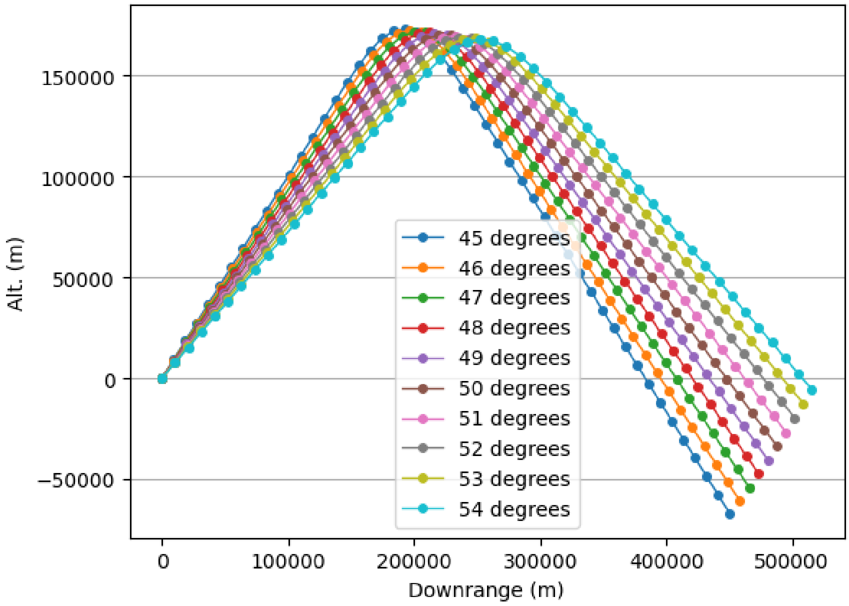

Figure 8: Simulation of Ray Propagation Across Various Angles in the Ionosphere

Launch Angle (Degrees)	**X Value at Ground Intersection**	**Total Traveling Length**	**Total Traveling Time**
45	386080.31	616467.91	0.00205489
46	392759.54	617140.09	0.00205713
47	408826.73	617805.11	0.00205935
48	425078.85	618462.70	0.00206154
49	431693.89	619112.58	0.00206371
50	448136.00	619754.49	0.00206585
51	464733.29	620388.22	0.00206796
52	481474.58	621013.52	0.00207005
53	487966.30	621630.19	0.00207210
54	504826.61	622238.02	0.00207413

Table 1: Ground Intersection, Traveling Length, and Time for Different Launch Angles

To align the simulation data with real-world observations, the research introduced a layer of complexity by creating synthetic actual data. This was achieved by incorporating noise into the simulation data, mimicking the natural inconsistencies that would be encountered in a real-world scenario.

The optimization procedure aimed to exploit the capabilities of JAX's 'grad' function to automatically differentiate the loss function of the model. However, the implementation confronted a critical obstacle: the 'grad' function is designed for continuous functions, while the ionospheric model, solved by diffrax, was essentially discrete. Thus, the application of automatic differentiation to optimize the model proved to be unfeasible within the current setup, showcasing an intriguing paradox between the two approaches.

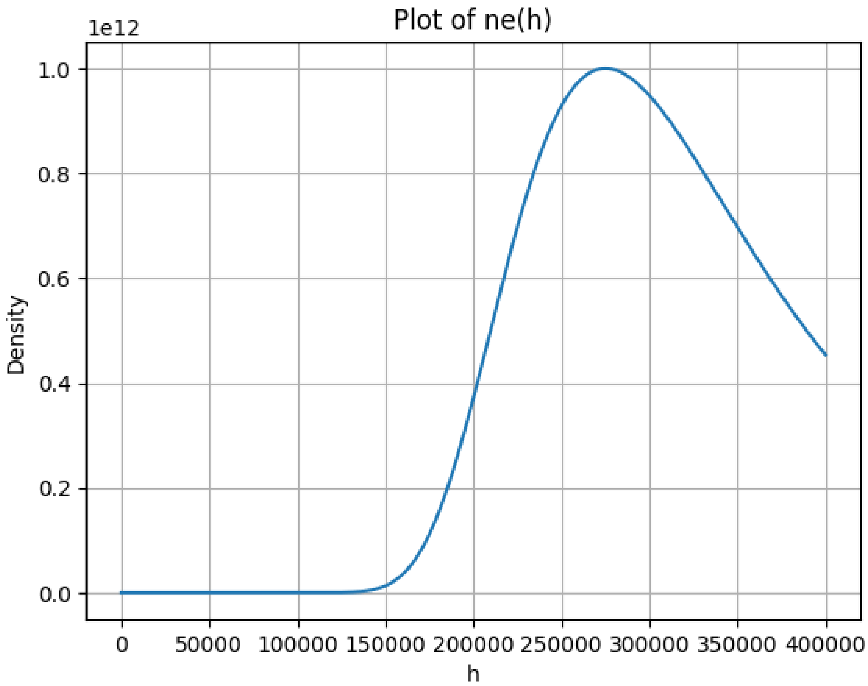

Figure 9: Visualization of the Continuous Electron Density Model *ne(h)* in the Ionosphere.

In prior research, the optimization problem was tackled using a gradient descent method applied to the discrete model. However, the current endeavor aimed to incorporate automatic differentiation into this procedure, which was identified as a challenge requiring further exploration.

While there are alternative optimization methods available, they primarily offer approximations rather than exact solutions. The intricate intertwining of continuous and discrete elements in the model necessitates a sophisticated approach that accommodates both. The challenges encountered in this research underscore the complexities inherent in high-precision scientific computing, particularly when blending techniques from different mathematical frameworks.

Future endeavors could possibly seek to address this incompatibility and explore how automatic differentiation could be integrated with discrete model optimization, promising an exciting avenue of investigation in the field.

8 Conclusion and Future Directions

The pursuit of this research project has uncovered many intriguing aspects and challenges within the field of ionospheric research, particularly regarding the application of automatic differentiation tools for optimizing ionospheric model parameters. Although we have effectively employed JAX for computing derivatives, its integration with discrete models was found to be challenging, showcasing the apparent mismatch between automatic differentiation methods and discrete models.

The complexity of the task at hand, however, underscores the adaptability and effectiveness of the methods employed in this research. By applying the variational (forward) method, valuable insights were drawn regarding the evolution of state vectors. Additionally, the successful transformation of a continuous ionospheric model into a discrete format was achieved using a strategic combination of Diffrax and JAX. Notably, the integration of simulation data with real-world data was achieved by generating ray trajectories and subsequently calculating their corresponding travel times. This process not only validated the accuracy of the ionospheric model but also brought to light the intricacies of working with real-world data, emphasizing the necessity to account for variables such as noise and variations in parameters such as the speed of light due to the ionosphere's inhomogeneity.

This research has illuminated the considerable potential of automatic differentiation tools such as JAX in facilitating complex computations. However, it also underscores the need for further research aimed at reconciling the gap between automatic differentiation and discrete models. Future work could explore alternative optimization methods or investigate ways to make discrete models compatible with automatic differentiation. Moreover, accounting for variations in the speed of light due to the inhomogeneous electron density within the ionosphere could further enhance the precision of future models.

In conclusion, this research project has broadened our understanding of ionospheric modeling, simultaneously highlighting new avenues for exploration within the realm of automatic differentiation and its application to other complex scientific problems.

References

[1] Diffrax ordinary differential equations (ODEs). *Diffrax*. Available at: https://docs.kidger.site/diffrax/usage/getting-started/#ordinary-differential-equations-odes.

[2] Wave. *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Wave.

[3] JAX, taking derivatives with grad(). *JAX*. Available at: https://jax.readthedocs.io/en/latest/notebooks/quickstart.html.

[4] Automatic differentiation. *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Automatic_differentiation.

[5] Zhengqing Yun, and Magdy F. Iskander. "Ray tracing for radio propagation modeling: Principles and applications." *IEEE access*, 3, (2015): 1089-1100.

[6] The Seismic Wave Equation. *IGPP*. Available at: https://igppweb.ucsd.edu/~guy/sio227a/ch3.pdf.

[7] Google Colab. *Google Collaboratory*, 2023. URL https://colab.research.google.com.

[8] Wikipedia. *Ionosphere*, 2023. URL https://en.wikipedia.org/wiki/Ionosphere.

[9] Smart, J.T. and Alonso, J.J. (2021). *CPU Parallelization and GPU Acceleration of SUAVE: Advancements in Sampling and Optimization*. In *AIAA Scitech 2021 Forum*, 1356.