

Ionosphere Ray-Tracing of RF Signals and Solution Sensitivities to Model Parameters

Mark L. Psiaki

Virginia Tech, Blacksburg, VA 24061, USA

BIOGRAPHY

Mark L. Psiaki is Professor and Kevin Crofton Faculty Chair of Aerospace and Ocean Engineering at Virginia Tech. He is also Professor Emeritus of Mechanical and Aerospace Engineering at Cornell University. He holds a Ph.D. in Mechanical and Aerospace Engineering from Princeton University. His research interests are in the areas of GNSS technology and applications, remote sensing, spacecraft attitude and orbit determination, and general estimation, filtering, and detection.

ABSTRACT

Methods are developed to determine the refracted propagation paths of high-frequency RF signals in the ionosphere and to determine the sensitivities of these paths to changes of the input model parameters. These techniques are being developed to support the assimilation of data from mono-static and multi-static ionosondes with the goal of improving parameterized estimates of ionosphere electron density profiles. An additional application area is that of navigation using signals from a ground-based network of high-frequency beacons. A nonlinear two-point boundary value problem solver is developed using the shooting method with Newton updates. Robust convergence is achieved by seeding the algorithm with carefully designed first guesses of the wave vector's initial aiming angles and terminal group delay. Partial derivative sensitivities of the ray-path solution are calculated using the adjoint of the two-point boundary value problem. This approach speeds the calculations when the partial derivatives of many ionosphere model parameters need to be computed. The new algorithm has been applied successfully to determine the paths of O-mode and X-mode radio waves between known transmitter and receiver locations and to spitze reflection points. Spitzze singularities pose no difficulties for the new algorithm because it uses a Hamiltonian ray-path formulation that remains non-singular at a spitze.

INTRODUCTION

Several areas of current scientific and engineering interest require an ability to calculate the ray paths of high-frequency (HF) radio signals through an ionosphere model. They also require an ability to determine how model changes affect the ray paths. One application is the fusion of ionosonde and GPS data to develop a local ionosphere model [1]. Another is the joint estimation of receiver position and ionosphere model corrections based solely on the observables of signals received from a ground-based network of HF beacons [2]. A third application is the assimilation of data from a network of HF beacons/receivers into a regional ionosphere model [3].

The latter two application areas are illustrated in Fig. 1. Two high-frequency beacons are shown as blue dots, and a receiver is shown as a red dot. A third beacon is out of view over the horizon in eastern Canada. Three refracting ray paths are shown between the 3 beacons and the receiver, one in green, one in purple, and one in tan. Two of the ray paths undergo multiple hops, but the purple one undergoes just a single hop between the transmitter and the receiver. Reference [2] proposes to use observables from these ray paths to estimate the receiver position, the receiver clock offset, and corrections to a parameterized model of the ionosphere electron density profile $N_e(\vec{r}; \mathbf{p})$, where \vec{r} is the Cartesian position vector along the ray path and \mathbf{p} is a vector of parameters that characterize the profile. The system discussed in [3] assumes that the receiver location is known and only tries to estimation ionosphere model corrections. In this latter system, the network of HF beacons/receivers constitutes a sort of multi-static ionosonde.

This paper has two main objectives. The first one is to calculate the observables of a HF ray path that refracts through the ionosphere by solving the ray-tracing equations. These observables include the group delay, the carrier phase, and the arrival and departure directions of the signal at the two ends of the ray path. Two types of ray paths are considered. One type undergoes total reflection in the ionosphere so that the transmitter and the receiver are collocated. This is the ionosonde ray-

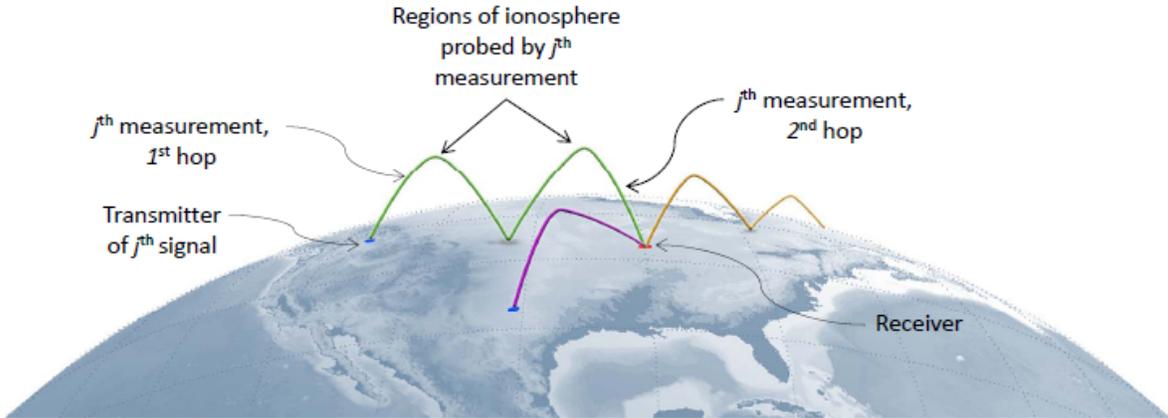


Fig. 1: High-Frequency beacons, a receiver, and representative ray paths that refract off of the ionosphere.

path problem. The second type of ray path is a point-to-point path from a known transmitter location to a different known receiver location. Ray paths of this second type can be concatenated to produce multi-hop paths. The explicit calculation of multi-hop paths is beyond the scope of this paper, except that this paper's developments for single-hop paths will provide all of the needed outputs for use in a concatenated multi-hop calculation.

This paper's second main objective is to develop a means of computing the partial derivative sensitivities of the ray-path observables to problem inputs. For ionosonde reflection ray paths, the needed sensitivities are the partial derivatives with respect to the elements of \mathbf{p} , which define the ionosphere electron density profile $N_e(\vec{r}; \mathbf{p})$. For point-to-point ray paths, additional calculations are developed for the partial derivative sensitivities with respect to the transmitter and receiver locations.

This paper makes four contributions to the technique of RF ray-tracing. The first is a modified pair of Hamiltonians, different from those given in [4], for use in defining the Hamiltonian differential equations of the ray path. The second contribution is a spline method for transitioning from the differential equations that are based on the first Hamiltonian to the differential equations that are based on the second Hamiltonian. The first Hamiltonian applies near free space, and the second applies in regions of high electron density, i.e., near a reflection point/spitze. The third contribution is a robust solver for the ray-tracing nonlinear Two-Point Boundary Value Problem (TPBVP). It uses the shooting method and Newton's method. It iteratively determines the initial wave vector direction and the final group delay that cause the ray-path to satisfy the terminal boundary conditions. This ray-path solver seeds Newton's method with good first guesses that it generates

using a simplified ray-path model. The fourth contribution is an adjoint-based method for calculating the sensitivities of various ray-path observables to problem parameters. The calculated sensitivities are partial derivatives of the TPBVP solution observables taken with respect to input quantities that define the TPBVP.

The remainder of this paper is divided into 4 main sections plus conclusions. Section II defines the 2 new Hamiltonians, and it uses them to develop the ray-tracing differential equations. This section also presents the two sets of boundary conditions for the two classes of ray-tracing problems that are dealt with in this paper. Section III describes the solution method for the ray-tracing nonlinear TPBVP. It develops a combined shooting-method/Newton's-method. It includes a technique for generating good first guesses to seed the shooting/Newton algorithm. Section IV derives methods for determining the partial derivative sensitivities of the ray-tracing solution and its observables with respect to problem inputs that include ionosphere parameters and the initial and final locations of the ray path. Section V presents example ray-tracing solutions. Section VI summarizes this paper's contributions and gives its conclusions.

II. HAMILTONIAN FORMULATION OF RAY-TRACING PROBLEM

This section formulates the differential equations and the boundary conditions of two ray-tracing problems. One is an ionosonde reflection problem, and the other is a point-to-point problem.

A. Hamiltonian Differential Equations

A ray-tracing solution can be characterized by the time histories of its Cartesian position vector \vec{r} and its Cartesian wave vector \vec{k} . They obey the Hamiltonian differential equations [4]:

$$\begin{aligned}\frac{d\vec{r}}{dP'} &= -\frac{\left(\frac{\partial H}{\partial \vec{k}}\right)^T}{\frac{\partial H}{\partial k_0}} \\ \frac{d\vec{k}}{dP'} &= \frac{\left(\frac{\partial H}{\partial \vec{r}}\right)^T}{\frac{\partial H}{\partial k_0}}\end{aligned}\quad (1)$$

where H is an appropriate Hamiltonian, $P' = ct_{group}$ is the range-equivalent group delay, and $k_0 = \omega/c$ is the free-space wave number with ω being the transmission frequency. The transpose operations in these equations compensate for the fact that the two gradients of the scalar Hamiltonian H with respect to the column vectors \vec{r} and \vec{k} are defined to be row vectors.

The same Hamiltonian can be used to develop a differential equation for the range-equivalent carrier phase $P = \phi/k_0$ with ϕ being the carrier phase in radians. This differential equation takes the form

$$\frac{dP}{dP'} = -\frac{\left(\frac{\partial H}{\partial \vec{k}}\right)^T \vec{k}}{\frac{\partial H}{\partial k_0} k_0}\quad (2)$$

The pair of differential equations in Eq. (1) can be used to develop the following state-space form of the ray-tracing differential equations

$$\frac{d\mathbf{x}}{d\tau} = P'_f \frac{d\mathbf{x}}{dP'} = \mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})\quad (3)$$

where the state vector is

$$\mathbf{x} = \begin{bmatrix} \vec{r}/P'_f \\ \vec{k}/k_0 \end{bmatrix}\quad (4)$$

and where P'_f is the total group delay from the initial time to the final time along the ray path. The independent variable $\tau = P'/P'_f$ is the fractional group delay along the ray path. It is non-dimensional and ranges from $\tau = 0$ at the start of the ray path to $\tau = 1$ at the end. The 6-element state vector \mathbf{x} is non-dimensional, as is the 6-element vector function $\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})$. This function can be derived from the right-hand sides in Eq. (1).

The range-equivalent carrier phase can be appended to the state vector in order to develop a 7-element state as follows:

$$\mathbf{x} = \begin{bmatrix} \vec{r}/P'_f \\ \vec{k}/k_0 \\ P/P'_f \end{bmatrix}\quad (5)$$

If using this latter state definition, then the $\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})$ function in the state differential equation must be augmented to become a 7-element function by including a final term that is based on the scalar carrier-phase differential equation in Eq. (2). This augmented 7-element state is not needed in order to solve for a ray path, but it is needed if one wants to compute the carrier-phase observable.

B. Two New Appleton-Hartree Hamiltonians

The ray-tracing differential equations use Hamiltonians that are derived from Maxwell's equations for radio wave propagation in a lossy magnetoplasma. They assume that the wavelength is vanishingly small relative to the spatial scales of the variations of the electron density, the Earth's magnetic field, and the electron-neutral collision frequency. This is the same analysis that has been used to derive the Appleton-Hartree index of refraction [5]. The ray-tracing solution ensures that $\|\vec{k}\|/k_0$ equals the Appleton-Hartree index of refraction in the lossless case.

Reference [4] presents several Hamiltonians that are based on the Appleton-Hartree index of refraction or on the underlying analysis that produces it. The following new Hamiltonian is closely related to the one defined in Eq. (21) of [4]:

$$\begin{aligned}H_1(\vec{r}, \vec{k}, k_0, \mathbf{p}) &= \\ &= \frac{1}{2} \text{real} \left\{ \left[\left(\frac{\vec{k}^T \vec{k}}{k_0^2} \right) - n_{AH}^2(\vec{r}, \frac{\vec{k}}{\|\vec{k}\|}, k_0, \mathbf{p}) \right] \right. \\ &\quad \left. \times [(1 - jZ)^2 - Y^2](1 - jZ) \right\}\end{aligned}\quad (6)$$

where n_{AH} is the lossy Appleton-Hartree index of refraction [5], $Y = \omega_{eg}/\omega$ is the ratio of the unsigned electron gyro frequency to the transmission frequency, and $Z = \nu/\omega$ is the ratio of the electron-neutral collision frequency to the transmission frequency. The final two factors on the right-hand side of this equation, the terms that involve Y and jZ , make this Hamiltonian different from that in Eq. (21) of [4]. They have been introduced in order to make this Hamiltonian similar the second one that is used in this paper's developments.

Similar to its counterpart in [4], the Hamiltonian in Eq. (6) is used near free space, where the electron density $N_e(\vec{r}; \mathbf{p})$ is relatively small. It is not appropriate near an

ionosonde reflection point/spitze singularity. This Hamiltonian has singularities at such points. Therefore a second Hamiltonian is used near such points, exactly as in [4].

This paper's new Hamiltonian for use near a spitze is significantly different from the corresponding Hamiltonian given in Eq. (22) of [4]. The new Hamiltonian equals one half the real part of an eigenvalue of a 3×3 matrix. This matrix constitutes the coefficient that multiplies the electric field vector in a homogeneous equation that is derived using Maxwell's equations. For a given direction of the wave vector, the square of the Appleton-Hartree index of refraction can be determined by solving a generalized eigenvalue problem for the matrix. This matrix is

$$D(\vec{r}, \vec{k}, k_0, \mathbf{p}) = \left[\left(\frac{\vec{k}^T \vec{k}}{k_0^2} - 1 \right) I - \frac{\vec{k} \vec{k}^T}{k_0^2} \right] \times [(1 - jZ)^2 - Y^2](1 - jZ) + X \{ (1 - jZ)^2 I - Y^2 \hat{\mathbf{B}}_0 \hat{\mathbf{B}}_0^T - j(1 - jZ) Y [\hat{\mathbf{B}}_0 \times] \} \quad (7)$$

where $X = \omega_p^2/\omega^2$ is the square of the ratio of the plasma frequency to the transmission frequency and where $\hat{\mathbf{B}}_0 = \vec{\mathbf{B}}_0/|\vec{\mathbf{B}}_0|$ is the unit vector that points in the direction of the ambient geomagnetic field vector $\vec{\mathbf{B}}_0$. The notation $[\hat{\mathbf{B}}_0 \times]$ indicates the 3×3 skew-symmetric cross-product-equivalent matrix. It produces the same result when multiplying a vector on the left as would be produced by taking the cross product of that vector with $\hat{\mathbf{B}}_0$ on the left.

The second Hamiltonian takes the form:

$$H_2(\vec{r}, \vec{k}, k_0, \mathbf{p}) = \frac{1}{2} \text{real}(\lambda) \quad (8)$$

where λ is an eigenvalue of the D matrix:

$$\left[D(\vec{r}, \vec{k}, k_0, \mathbf{p}) - \lambda I \right] \vec{\mathbf{E}} = 0 \quad (9)$$

and $\vec{\mathbf{E}}$ is the associated electric field eigenvector

This Hamiltonian does not experience any singularities at a reflection point/spitze. Therefore, it is useful when the electron density is high. This Hamiltonian is not useful when the electron density is small because of difficulties in distinguishing the correct eigenvalue/eigenvector pair to use of the 3 that exist.

The two Hamiltonians have different methods of enforcing selectivity for the O mode or the X mode of radio wave propagation. For H_1 , this selection is enforced by the choice of a sign in the Appleton-Hartree formula for n_{AH}^2 [5]. For H_2 , this selection is enforced via proper choice of the eigenvalue/eigenvector pair. In the present context, this choice is made by enforcing continuity of the eigenvector

direction $\vec{\mathbf{E}}$ along the ray path. Abrupt changes are ruled out. The initial selection of an appropriate direction at the change from using H_1 to using H_2 is accomplished by computing the same eigenvector from its corresponding Appleton-Hartree formula [5] whenever H_1 is being used. When enforcing continuity, one must be careful to compute a sensible directional difference between two neighboring complex-valued $\vec{\mathbf{E}}$ vectors along a given ray path.

The new H_2 Hamiltonian is conjectured to be more reliable at maintaining the proper mode selection than the one in Eq. (22) of [4]. The latter Hamiltonian has no explicit mode selection mechanism. It relies on differential equation solution continuity to select the correct wave propagation mode. The new H_2 Hamiltonian relies on a modified version of continuity, that of its associated electric field eigenvector. It is conjectured that this form of continuity provides stronger guarantees of proper mode selection.

The dependence of H_2 on the position vector \vec{r} and the ionosphere parameter vector \mathbf{p} comes partly through the dependencies of X , Y , and Z on these quantities. These dependencies take the forms:

$$\begin{aligned} X &= \frac{N_e(\vec{r}; \mathbf{p}) q_e^2}{m_e \varepsilon_0 \omega^2} \\ Y &= \frac{|\vec{\mathbf{B}}_0(\vec{r})| |q_e|}{m_e \omega} \\ Z &= \frac{\nu(\vec{r})}{\omega} \end{aligned} \quad (10)$$

where q_e is the electron charge, m_e is the electron mass, ε_0 is the permittivity of free space, and $\nu(\vec{r})$ is the electron-neutral collision frequency. Additional H_2 dependence on \vec{r} comes through the dependence of $\hat{\mathbf{B}}_0$ on \vec{r} . The Earth magnetic field model $\vec{\mathbf{B}}_0(\vec{r})$ used in the present study is the IGRF field [6].

One must calculate partial derivatives of an eigenvalue of the $D(\vec{r}, \vec{k}, k_0, \mathbf{p})$ matrix in order to compute the partial derivatives of H_2 that are needed in Eqs. (1) and (2). The following linear system of equations can be solved in order to compute the partial derivatives of the eigenvalue λ and the associated eigenvector $\vec{\mathbf{E}}$ with respect to the quantity η :

$$\begin{bmatrix} (D - \lambda I) & -\vec{\mathbf{E}} \\ \vec{\mathbf{E}}^{*T} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \vec{\mathbf{E}}}{\partial \eta} \\ \frac{\partial \lambda}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -\frac{\partial D}{\partial \eta} \vec{\mathbf{E}} \\ 0 \end{bmatrix} \quad (11)$$

The notation $()^{*T}$ indicates the complex conjugate of the transpose of the vector in question. The scalar differentiation variable η is any quantity on which D depends,

i.e., any component of \vec{r} , any component of \vec{k} , k_0 , or any component of \mathbf{p} . The second line in this equation ensures that the (possibly complex-valued) \vec{E} vector does not change its norm and that it does not undergo any unnecessary simultaneous complex phase rotations of all of its elements.

This eigenvalue differentiation technique breaks down if λ is a repeated eigenvalue of the D matrix. Repeated or nearly repeated eigenvalues occur near free space, when X is small. That is why Hamiltonian H_2 is used only in regions of high X , which correspond to regions of high electron density $N_e(\vec{r}; \mathbf{p})$.

C. Transition Between the Two Hamiltonians

The Hamiltonians H_1 and H_2 yield identical ray-tracing differential equations in the lossless $Z = 0$ case. The trailing factors that involve Y and jZ in the H_1 Hamiltonian of Eq. (6) have been included in hopes of making the two Hamiltonians' differential equations more similar in the lossy case of nonzero Z . The following rationale inspired the inclusion of these terms: In effect, these same factors multiply the $D(\vec{r}, \vec{k}, k_0, \mathbf{p})$ matrix of Eq. (7). The net effect is that the D matrix eigenvalues are multiplied by these same factors, which makes them similar to the formula for H_1 in Eq. (6).

Nevertheless, these two Hamiltonians will yield slightly different $\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})$ functions in Eq. (3). Let them be designated as $\mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p})$ and $\mathbf{f}_2(\mathbf{x}, P'_f, \mathbf{p})$. It is necessary to transition from \mathbf{f}_1 to \mathbf{f}_2 as X increases along a ray path. The transition method used in [4] is to switch abruptly at $X = 0.1$.

The transition used here is different. It takes the gradual form:

$$\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p}) = \begin{cases} \mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p}) & \text{if } X \leq X_1 \\ \mathbf{f}_T(\mathbf{x}, P'_f, \mathbf{p}, X) & \text{if } X_1 < X < X_2 \\ \mathbf{f}_2(\mathbf{x}, P'_f, \mathbf{p}) & \text{if } X_2 \leq X \end{cases} \quad (12)$$

where

$$\mathbf{f}_T(\mathbf{x}, P'_f, \mathbf{p}, X) = \mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p}) + s(X)[\mathbf{f}_2(\mathbf{x}, P'_f, \mathbf{p}) - \mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p})] \quad (13)$$

with

$$s(X) = \left(\frac{X - X_1}{X_2 - X_1} \right)^2 \left[3 - 2 \left(\frac{X - X_1}{X_2 - X_1} \right) \right] \quad (14)$$

Equations (12)-(14) implement a cubic spline transition from using $\mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p})$ below X_1 to using $\mathbf{f}_2(\mathbf{x}, P'_f, \mathbf{p})$ above X_2 . The cubic spline function $S(X)$ implements

this transition. It changes smoothly from 0 to 1 over the interval from X_1 to X_2 , and it has zero-valued derivatives at X_1 and X_2 . The resulting composite $\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})$ function in Eq. (12) is continuous with continuous first partial derivatives with respect to \mathbf{x} if each of its constituent functions $\mathbf{f}_1(\mathbf{x}, P'_f, \mathbf{p})$ and $\mathbf{f}_2(\mathbf{x}, P'_f, \mathbf{p})$ are similarly smooth.

The X transition range limits X_1 and X_2 are user selectable. They must obey $X_1 < X_2$. For an O-mode ray path they must be less than 1. Typical values might be $X_1 = 0.4$ and $X_2 = 0.6$. For an X-mode ray path, the allowable values of X_1 and X_2 depend on the magnitude of Y . Consider the case $Y > 1$, i.e., the case where the transmission frequency is lower than the electron gyro frequency. X_1 and X_2 both must be less than 1 in this situation, and the values $X_1 = 0.4$ and $X_2 = 0.6$ are reasonable. For $Y < 1$, the transition range limits must be less than the reflection value $X_{refl} = 1 - Y$. Typical values used in this study are $X_1 = 0.4(1 - Y)$ and $X_2 = 0.6(1 - Y)$ when calculating an X-mode ray path with $Y < 1$.

D. Ray-Tracing Boundary Conditions

Each of the two ray-tracing problems has initial and terminal boundary conditions. There are 4 scalar initial boundary conditions. They apply at $\tau = 0$ and take the form

$$\begin{aligned} 0 &= [I, 0]\mathbf{x}(0) - \frac{\vec{r}_0}{P'_f} \\ 0 &= H_1(\vec{r}_0, \vec{k}(0), k_0, \mathbf{p}) \\ &= H_1(\vec{r}_0, \{[0, k_0 I]\mathbf{x}(0)\}, k_0, \mathbf{p}) \end{aligned} \quad (15)$$

The first line of this equation sets the transmitter position at \vec{r}_0 . The second line ensures that the initial wave vector has magnitude equal to the Appleton-Hartree index of refraction multiplied by k_0 . After initialization at 0, the Hamiltonian will remain zero due to the form of the Hamiltonian differential equations in Eq. (1).

There are 3 scalar termination boundary conditions at $\tau = 1$. For the ionosonde reflection case, these conditions take the form:

$$0 = \vec{k}(1)/k_0 = [0, I]\mathbf{x}(1) \quad (16)$$

The terminal boundary conditions for the point-to-point case take the form

$$0 = [I, 0]\mathbf{x}(1) - \frac{\vec{r}_f}{P'_f} \quad (17)$$

where \vec{r}_f is the known receiver location at the end of the ray path.

The terminal boundary conditions for both cases can be written in the general form:

$$0 = A_f \mathbf{x}(1) + \frac{\mathbf{b}_f}{P'_f} \quad (18)$$

where A_f is a 3×6 matrix and \mathbf{b}_f is a 3×1 vector. This form of the terminal boundary conditions will be used to develop the ray-tracing solution algorithm in Section III.

The initial and terminal boundary conditions can be rewritten in the following general forms:

$$\begin{aligned} 0 &= \mathbf{g}_0[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0] \\ 0 &= \mathbf{g}_f[\mathbf{x}(1), P'_f, \vec{\mathbf{r}}_f] \end{aligned} \quad (19)$$

with the understanding that \mathbf{g}_f will be independent of $\vec{\mathbf{r}}_f$ for the ionosonde reflection case of Eq. (16). These forms of the boundary conditions will be used to develop the ray-tracing sensitivity calculations in Section IV.

III. RAY-TRACING SOLUTION METHOD

The ray-tracing problem is a nonlinear TPBVP. There is no closed-form solution for the general case. Therefore, numerical methods have been developed to solve this problem. The main solution algorithm combines the shooting method to deal with the terminal boundary conditions and Newton's method to deal with the nonlinearities. The shooting method works by guessing the free initial conditions, numerically integrating forward to the terminal condition, checking whether the terminal conditions are solved, incrementing the guess of the free initial condition components in a way that will reduce the error in the terminal boundary conditions, and iterating this process until the free initial state vector components converge to values that cause the terminal boundary conditions to be satisfied. Newton's method provides a means of computing the needed increments to the free components of the initial state vector. It converges very rapidly when it gets near the solution.

Newton's method can fail to converge if its guess starts too far from the solution. Two strategies are employed to avoid this mode of solution failure. The first strategy is to develop a reasonably good first guess of the free initial state vector components. The second strategy is to employ the Levenberg-Marquardt technique [7] as a means of forcing the Newton algorithm to converge to a local minimum of the sum of the squares of the errors in the terminal boundary conditions.

A. Newton's Method Implementation of the Shooting Algorithm

The Newton's method implementation of the shooting algorithm for the TPBVP solves 3 equations in 3 unknowns. The 3 unknowns consist of the 2 free parameters of the unknown direction of the initial wave vector $\hat{\mathbf{s}} = \vec{\mathbf{k}}(0)/\|\vec{\mathbf{k}}(0)\|$ and the unknown range-equivalent terminal group delay P'_f . The following algorithm implements a Newton/shooting method for refining initial guesses of $\hat{\mathbf{s}}$ and P'_f until the terminal boundary conditions are satisfied:

- 1) Obtain initial guesses of $\hat{\mathbf{s}}$ and P'_f .
- 2) Solve for the value of the initial wave vector magnitude k_{init} that satisfies the initial condition on the Hamiltonian $0 = H_1(\vec{\mathbf{r}}_0, k_{init}\hat{\mathbf{s}}, k_0, \mathbf{p})$. This equation is quadratic in k_{init} and can be solved in closed form.
- 3) Set the initial ray-tracing state vector to

$$\mathbf{x}(0) = \begin{bmatrix} \vec{\mathbf{r}}_0/P'_f \\ (k_{init}/k_0)\hat{\mathbf{s}} \end{bmatrix} \quad (20)$$

- 4) Numerically integrate the 6-element form of the ray-tracing state differential equation in Eq. (3) from $\tau = 0$ to $\tau = 1$. Use a Runge-Kutta method such as the Dormand-Prince 4/5 method [8]. Use predefined step sizes here. Adaptation will take place elsewhere.
- 5) Check whether the norm of the terminal boundary condition $\|A_f \mathbf{x}(1) + \mathbf{b}_f/P'_f\|$ is sufficiently small. If so, then stop. Otherwise, proceed to the next step.
- 6) Compute two allowable perturbation directions for $(k_{init}/k_0)\hat{\mathbf{s}}$ by determining $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ such that $[\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3]$ is an orthonormal triad with

$$\hat{\mathbf{v}}_3 = \frac{\left(\frac{\partial H_1}{\partial \vec{\mathbf{k}}} \Big|_{(\vec{\mathbf{r}}_0, k_{init}\hat{\mathbf{s}}, k_0, \mathbf{p})} \right)^T}{\left\| \frac{\partial H_1}{\partial \vec{\mathbf{k}}} \Big|_{(\vec{\mathbf{r}}_0, k_{init}\hat{\mathbf{s}}, k_0, \mathbf{p})} \right\|} \quad (21)$$

$\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ can be determined by computing an orthonormal/upper-triangular factorization of $\hat{\mathbf{v}}_3$ [7].

- 7) Initialize the 6×3 linearized solution sensitivity matrix

$$\Gamma(0) = \begin{bmatrix} 0 & 0 & -\vec{\mathbf{r}}_0/(P'_f)^2 \\ \hat{\mathbf{v}}_1 & \hat{\mathbf{v}}_2 & 0 \end{bmatrix} \quad (22)$$

- 8) Numerically integrate the following linear sensitivity matrix ordinary differential equation from $\tau = 0$ to $\tau = 1$

$$\frac{d\Gamma(\tau)}{d\tau} = F(\tau)\Gamma(\tau) + [0, 0, \mathbf{u}(\tau)] \quad (23)$$

where

$$\begin{aligned} F(\tau) &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{[\mathbf{x}(\tau), P'_f, \mathbf{p}]} \\ \mathbf{u}(\tau) &= \left. \frac{\partial \mathbf{f}}{\partial P'_f} \right|_{[\mathbf{x}(\tau), P'_f, \mathbf{p}]} \end{aligned} \quad (24)$$

This matrix differential equation must be numerically integrated using the same Runge-Kutta method as is used in Step 4. It must use identical steps, and it must be integrated in a manner that is identical to what would occur had it been integrated simultaneously with $\mathbf{x}(\tau)$ in Step 4 as though \mathbf{x} and the 3 columns of Γ formed one large state vector with 24 elements.

- 9) Solve the following linearized terminal boundary condition for the initial wave vector increment parameters $\Delta\alpha_1$ and $\Delta\alpha_2$ and for the terminal group delay increment $\Delta P'_f$:

$$0 = A_{bc} \begin{bmatrix} \Delta\alpha_1 \\ \Delta\alpha_2 \\ \Delta P'_f \end{bmatrix} + \mathbf{b}_{bc} \quad (25)$$

where

$$\begin{aligned} A_{bc} &= \left\{ A_f \Gamma(1) + \left[0, 0, -\frac{\mathbf{b}_f}{(P'_f)^2} \right] \right\} \\ \mathbf{b}_{bc} &= A_f \mathbf{x}(1) + \frac{\mathbf{b}_f}{P'_f} \end{aligned} \quad (26)$$

- 10) Determine a new initial direction of the wave vector and a new terminal group delay:

$$\begin{aligned} \hat{\mathbf{s}}_{new} &= \frac{(k_{init}/k_0)\hat{\mathbf{s}} + \hat{\mathbf{v}}_1 \Delta\alpha_1 + \hat{\mathbf{v}}_2 \Delta\alpha_2}{\|(k_{init}/k_0)\hat{\mathbf{s}} + \hat{\mathbf{v}}_1 \Delta\alpha_1 + \hat{\mathbf{v}}_2 \Delta\alpha_2\|} \\ P'_{fnew} &= P'_f + \Delta P'_f \end{aligned} \quad (27)$$

- 11) Replace $\hat{\mathbf{s}}$ by $\hat{\mathbf{s}}_{new}$ and P'_f by P'_{fnew} and return to Step 2

The first partial derivative of \mathbf{f} with respect to \mathbf{x} in Eq. (24) requires second partial differentiation with respect to $\vec{\mathbf{r}}$ of the problem model functions $N_e(\vec{\mathbf{r}}; \mathbf{p})$, $\vec{\mathbf{B}}_0(\vec{\mathbf{r}})$, and $\nu(\vec{\mathbf{r}})$. This is true because the computation of \mathbf{f} requires first partial differentiation with respect to $\vec{\mathbf{r}}$ as per Eq. (1). Therefore, these second derivatives must exist and be continuous. One way to achieve the needed smoothness of $N_e(\vec{\mathbf{r}}; \mathbf{p})$ is to employ a model that works with smooth vertical profiles. The parameters of the vertical profiles are modeled as 2nd-order smooth functions of latitude and longitude through the use of a bi-quintic latitude/longitude spline. An example $N_e(\vec{\mathbf{r}}; \mathbf{p})$ profile of this type is discussed in [9].

B. Levenberg-Marquardt Modifications to Newton's Method

If the current guesses of $\hat{\mathbf{s}}$ and P'_f are far from the solution, then the terminal boundary condition error norm computed in Step 5 of the Newton algorithm may increase or remain the same relative to the previous step. It should decrease in order to ensure convergence to a solution.

A decrease in the Step-5 norm can be ensured through use of the Levenberg-Marquardt method for computing the solution increments [7]. The Levenberg-Marquardt method determines $\Delta\alpha_1$, $\Delta\alpha_2$, and $\Delta P'_f$ by a means that differs from the linear equation solution of the Newton algorithm's Step 9. They are determined using the formula:

$$\begin{bmatrix} \Delta\alpha_1 \\ \Delta\alpha_2 \\ \Delta P'_f \end{bmatrix} = -(A_{bc}^T A_{bc} + \gamma I)^{-1} A_{bc}^T \mathbf{b}_{bc} \quad (28)$$

for some non-negative value of the scalar Levenberg-Marquardt parameter γ . It is easy to verify that the Levenberg-Marquardt values for $\Delta\alpha_1$, $\Delta\alpha_2$, and $\Delta P'_f$ equal the Newton's-method values from Step 9 when $\gamma = 0$. For larger γ values the increments tend to decrease, and they become vanishingly small as $\gamma \rightarrow \infty$. It is not difficult to prove that there must exist a finite non-negative value γ such that the computed increments from Eq. (28) cause the terminal boundary condition error norm to decrease in Step 5 of the algorithm.

The Levenberg-Marquardt method can be incorporated into the Newton algorithm of the previous subsection as follows: Each major outer iteration loop from Steps 2 through 11 is augmented by an inner iteration loop that implements a tentative set of calculations from the Step-9 increment calculation to the subsequent Step-5 test of the terminal boundary condition error norm. If the error norm of Step 5 has not decreased relative to its value on the last major iteration, then the operations are repeated starting from the preceding Step 9, except that the Levenberg-Marquardt increment calculation in Eq. (28) is used in place of the solution of the linear system in Eq. (25), and this calculation uses an increased γ value.

The value $\gamma = 0$ is tried for the initial increment calculation of each major outer iteration in order to attempt a full Newton step. The full Newton step yields fast convergence near the solution, which is why it is preferred. If the initial Newton step fails to decrease the terminal boundary condition error, then an alternate set of $\Delta\alpha_1$, $\Delta\alpha_2$, and $\Delta P'_f$ increments are computed using a positive value of γ . This initial positive value is chosen to equal a small fraction of the square of the smallest singular value of the matrix A_{bc} or to equal an even smaller fraction of the square of the largest singular value of this matrix,

whichever is larger. If the resulting increments $\Delta\alpha_1$, $\Delta\alpha_2$, and $\Delta P'_f$ still fail to reduce the terminal boundary condition error norm at Step 5, then the current positive value of γ is increased through multiplication by a factor of 3. The steps from 9 through the subsequent Step 5 are again repeated. This inner iteration repeats itself until the terminal boundary condition error norm is decreased. At this point, the final successful value of γ may be larger than the first non-zero trial value by several factors of 3.

Finally, the values of \hat{s}_{new} and P'_{fnew} replace the original \hat{s} and P'_f values. Prior to achieving a decrease of the terminal boundary condition error, the proposed \hat{s}_{new} and P'_{fnew} are only accepted as tentative replacements for the original \hat{s} and P'_f from the previous major outer iteration of the algorithm.

C. Generation of a Ray-Tracing First Guess

It is important to supply a reasonably good first guess to the Newton's method implementation of the shooting algorithm. Even with the Levenberg-Marquardt modification, a good first guess is important. If the first guess is far from the solution, then the algorithm may require many iterations to converge. It may even converge to a non-zero local minimum of the terminal condition error norm, which is not a TPBVP solution.

The initial guess algorithm starts with knowledge of the ray path initial position \vec{r}_0 and the ray path terminal position \vec{r}_f . For the point-to-point case \vec{r}_f is a given of the problem. For the ionosonde reflection case \vec{r}_f is set equal to \vec{r}_0 .

The initial guess is generated in 3 stages. The first stage produces a coarse estimate of a peak "reflection" point along the ray path. Call it \vec{r}_r . Three conditions serve to determine this point. Before defining these conditions, it is helpful to define three unit direction vectors that all depend on \vec{r}_r :

$$\begin{aligned}\hat{s}_0 &= \frac{\vec{r}_r - \vec{r}_0}{\|\vec{r}_r - \vec{r}_0\|} \\ \hat{s}_f &= \frac{\vec{r}_f - \vec{r}_r}{\|\vec{r}_f - \vec{r}_r\|} \\ \hat{s}_{ra} &= \left(\frac{\partial h_{alt}}{\partial \vec{r}} \Big|_{\vec{r}_r} \right)^T\end{aligned}\quad (29)$$

where $h_{alt}(\vec{r})$ is the altitude of the Cartesian position vector \vec{r} above the WGS-84 Earth ellipsoid [10]. The unit direction vector \hat{s}_0 points along the initial ascending straight-line ray path approximation, \hat{s}_f points along the final descending straight-line ray path approximation, and \hat{s}_{ra} points in the local vertical direction at the approximate reflection point \vec{r}_r .

These three vectors can be used to develop the following system of 4 scalar equations in 4 unknowns:

$$\begin{aligned}0 &= \hat{s}_{ra} - \frac{\eta}{2}(\hat{s}_0 - \hat{s}_f) \\ 0 &= n_{AH}^2(\vec{r}_0, \hat{s}_0)[1 - (\hat{s}_{ra}^T \hat{s}_0)^2] - n_{AH}^2(\vec{r}_r, \hat{s}_0)\end{aligned}\quad (30)$$

The first of these equations is a 3×1 vector equation, and the last equation is a scalar equation. The four unknowns of this system are the three components of the reflection position \vec{r}_r and the scalar parameter η . These equations model total internal reflection at an abrupt change in refractive index. The vector equation assumes that the reflective surface is locally horizontal. It enforces the reflection conditions that \hat{s}_0 , \hat{s}_f , and \hat{s}_{ra} be coplanar and that the angle of incidence equal the angle of reflection. The scalar equation amounts to Snell's law for total internal reflection when the index of refraction changes abruptly from its value at \vec{r}_0 to its value at \vec{r}_r . All of this change is assumed to take place at \vec{r}_r . The Appleton-Hartree index of refraction formula used in Eq. (30) is the lossless version (i.e., $Z = 0$); it returns real values of n_{AH}^2 . Solution of this system of equations produces an initial coarse guess of \vec{r}_r .

The system of equations in Eq. (30) is nonlinear and must be solved iteratively. This nonlinear system of equations is also solved using Newton's method, but special precautions are taken to ensure that the solution converges robustly.

Initially a different set of equations is solved. It includes the first equation, but it replaces the scalar 2nd equation by the equation $h_r = h_{alt}(\vec{r}_r)$ for a pre-specified value of h_r . This amounts to solving for a proposed ray path with the known maximum altitude h_r rather than a ray path that satisfies Snell's law for total internal reflection. It is fairly easy to solve this alternate set of equations by using a spherical approximation of the Earth to generate a first guess and afterwards implementing a Newton's method solution starting from that guess. The result of this solution procedure is the candidate reflection point $\vec{r}_r(h_r)$.

Next, a grid of candidate h_r values is defined, and a set of candidate reflection points $\vec{r}_r(h_r)$ is generated for this grid. The right-hand side of the second equation in Eq. (30) is evaluated at these grid points, and the algorithm looks for the lowest-altitude grid points where the error in this latter equation changes sign. These two points are then used to seed a 1-dimensional guarded Newton's-method search along h_r for the point where the second equation in Eq. (30) is accurately satisfied.

The second stage of the ray path first-guess calculation solves an alternate system of equations. It is similar to the system in Eq. (30), except that it replaces the two

instances of \hat{s}_{ra} with a different reflection surface unit normal vector:

$$\hat{s}_{rb} = \frac{\left(\frac{\partial(n_{AH}^2)}{\partial \vec{r}} \Big|_{(\vec{r}_r, \hat{s}_0, k_0, \mathbf{p})} \right)^T}{\left\| \frac{\partial(n_{AH}^2)}{\partial \vec{r}} \Big|_{(\vec{r}_r, \hat{s}_0, k_0, \mathbf{p})} \right\|} \quad (31)$$

This is the unit normal to the surface of constant index of refraction that passes through the reflection point \vec{r} . This alternate system of equations discards the assumption that the abrupt change in index of refraction occurs normal to a constant-altitude surface. It makes the reasonable assumption that the change occurs normal to a surface of constant index of refraction. The resulting system of 4 equations in the 4-element vector of unknowns $[\vec{r}_r; \eta]$ is solved using Newton's method. The Newton iteration is seeded with the candidate reflection point that is the solution of the previous system in Eq. (30). This Newton iteration tends to converge rapidly because the first guess is very good for typical ionosphere models where the predominant electron density gradients occur along the vertical axis.

This second \vec{r}_r solution could be used to generate the required initial guesses for the Newton's method shooting algorithm of Subsection III-A. They take the form:

$$\hat{s} = \hat{s}_0$$

$$P'_f = \begin{cases} \|\vec{r}_r - \vec{r}_0\| & \text{ionosonde case} \\ \|\vec{r}_r - \vec{r}_0\| + \|\vec{r}_f - \vec{r}_r\| & \text{point-to-point case} \end{cases} \quad (32)$$

The third stage of the first-guess calculations starts by generating 5 candidate initial wave vector directions. They are

$$\begin{aligned} \hat{s}_a &= \hat{s}_0 \\ \hat{s}_b &= \cos(\theta)\hat{s}_0 + \sin(\theta)\hat{v}_1 \\ \hat{s}_c &= \cos(\theta)\hat{s}_0 - \sin(\theta)\hat{v}_1 \\ \hat{s}_d &= \cos(\theta)\hat{s}_0 + \sin(\theta)\hat{v}_2 \\ \hat{s}_e &= \cos(\theta)\hat{s}_0 - \sin(\theta)\hat{v}_2 \end{aligned} \quad (33)$$

where the unit direction vectors \hat{v}_1 and \hat{v}_2 are chosen so that $[\hat{v}_1, \hat{v}_2, \hat{s}_0]$ is an orthonormal triad. The aiming offset angle θ is set to 10 deg. These 5 candidate directions include and bracket the candidate initial wave vector direction \hat{s}_0 that has been generated in the previous stage of the initial guess calculations.

The final stage of the initial guess calculations numerically integrates the 6-state version of the ray-tracing system in Eq. (3). It uses the guessed value of P'_f from Eq. (32) in this integration, but it integrates twice as far as the nominal terminal time: It integrates from $\tau = 0$ out to $\tau = 2$. It then

performs a brute-force search for the τ value that solves the following optimization problem:

$$\begin{aligned} &\text{find: } \tau \\ &\text{to minimize: } J(\tau) = 0.5\|A_f \mathbf{x}(\tau) + \mathbf{b}_f/P'_f\|^2 \\ &\text{subject to: } 2/3 \leq \tau \leq 2 \end{aligned} \quad (34)$$

This optimization is used to refine the initial guess of P'_f by finding the point along the guessed ray path that comes closest to satisfying the terminal boundary conditions. If τ_{opt} is the optimal solution to the problem in Eq. (34), then

$$P'_{fnew} = P'_f \tau_{opt} \quad (35)$$

is a better initial guess of P'_f .

This calculation of ray paths and refinement of P'_f is carried out 5 times using the 5 different guesses of the initial wave vector direction defined in Eq. (33). Suppose that these 5 ray paths produce 5 different optimized values of the cost function in Eq. (34). Let them be designated as J_{opta} , J_{optb} , J_{optc} , J_{optd} , and J_{opte} . The chosen guess of \hat{s} will be the candidate from Eq. (33) that produces the lowest optimal terminal boundary condition error cost, and the chosen guess of P'_f will be the corresponding P'_{fnew} value from Eq. (35).

D. Numerical Integration Step Size Adaptation

It is advisable to implement step size adaptation when using a Runge-Kutta numerical integration algorithm. The Dormand-Prince 4/5 algorithm has been used in this study [8]. It is the same algorithm as is used in the popular MATLAB function ode45.m. It supports step size adaptation.

When using a numerical integration algorithm within an outer numerical solver for a TPBVP, however, step size adaptation can be harmful. It can cause a violation of the assumptions that go into the Newton's method linearization of the terminal boundary conditions in Eq. (25). Such violations can cause problems when trying to force the errors in the terminal boundary conditions to decrease for each major iteration of the algorithm of Subsections III-A and III-B.

The methods developed in this paper employ Dormand-Prince step size adaptation in two places. The first place is during the numerical integration calculations that are associated with the third stage of the initial guess generation procedure. These are the 5 integrations that start from the 5 candidate initial wave vector directions of Eq. (33). The sequence of Dormand-Prince Runge-Kutta step intervals associated with the best initial wave vector direction is then chosen as the fixed set of intervals for use in the Newton/shooting TPBVP solver. Suppose that the numerical integration grid points which have been used to

generate the best first guess are $\tau_0, \tau_1, \tau_2, \dots, \tau_k, \dots, \tau_N$. Note that $\tau_0 = 0$ and $\tau_N = 2$ by design of the trial integration and that $\tau_1, \dots, \tau_{N-1}$ are chosen by the step size adaptation algorithm. Suppose, also, that $\tau_k = \tau_{opt}$, the solution of the optimization problem in Eq. (34). Then the change from P'_f to P'_{fnew} in Eq. (35) necessitates the following change in the Dormand-Prince numerical integration grid points:

$$\tau_{inew} = \tau_i / \tau_k \text{ for all } i = 1, \dots, k \quad (36)$$

These are the numerical integration grid points that are used in the TPBVP solution.

It can be helpful to use two sets of numerical integration grid points in order to speed algorithm execution while achieving a specified level of numerical integration accuracy in the TPBVP solution. The idea is to do an initial TPBVP solution which uses a coarser set of numerical integration grid points that does not achieve the final desired level of integration accuracy. Afterwards, a second TPBVP solution is computed using a revised sequence of numerical integration grid points that will deliver the desired accuracy.

In this scenario, a coarse numerical integration error tolerance is used to generate the initial solution guesses described in the previous subsection along with the corresponding integration grid points defined in Eq. (36). These grid points are then used to solve the TPBVP. Afterwards, the TPBVP solution values of \hat{s} and P'_f are input to an adaptive step size Dormand-Prince numerical integration from $\tau = 0$ to $\tau = 1$. This second adaptive numerical integration uses the final target error tolerance to control its adaptation. The resulting numerical integration grid points are used as a fixed set of grid points in a second solution of the TPBVP. The first guesses of \hat{s} and P'_f for this second TPBVP solution are the final solution values from the problem that used the coarser numerical integration grid. Experience with this approach indicates that the second call of the TPBVP solver converges to a very accurate solution in a single Newton iteration.

IV. SENSITIVITIES OF THE RAY-TRACING SOLUTION AND ITS OBSERVABLES

Sometimes one needs to compute the partial derivatives of TPBVP solution outputs with respect to TPBVP inputs. The local ionosphere data assimilation problem of [1] and the regional ionosphere data assimilation problem of [3] could usefully employ the partial derivatives of group delay and carrier phase with respect to ionosphere parameters, i.e., $\partial P'_f / \partial \mathbf{p}$ and $\partial P(1) / \partial \mathbf{p}$. A joint navigation/ionosphere-correction algorithm like that proposed in [2] could use $\partial P'_f / \partial \mathbf{p}$ and the group delay

partial derivatives with respect to the receiver location $\partial P'_f / \partial \vec{\mathbf{r}}_f$. An algorithm that concatenates multiple ray-path hops needs the partial derivatives of the initial and final wave vectors with respect to the initial and final ray-path positions $\partial \vec{\mathbf{k}}(0) / \partial \vec{\mathbf{r}}_0$, $\partial \vec{\mathbf{k}}(1) / \partial \vec{\mathbf{r}}_0$, $\partial \vec{\mathbf{k}}(0) / \partial \vec{\mathbf{r}}_f$, and $\partial \vec{\mathbf{k}}(1) / \partial \vec{\mathbf{r}}_f$. This section develops methods for calculating these partial derivatives.

A. Direct Calculation of Partial Derivatives with Respect to Ionosphere Parameters

The direct method of calculating any given partial derivative starts by taking the partial derivative of the entire TPBVP. This includes differentiation of the ordinary differential equation in Eq. (3) and differentiation of the initial and terminal boundary conditions in Eq. (19).

Suppose that one wants to calculate partial derivatives with respect to the i^{th} element of the ionosphere parameter vector \mathbf{p} . Then the direct method results in the following linear TPBVP for the unknowns $\partial P'_f / \partial p_i$ and $\partial \mathbf{x}(\tau) / \partial p_i$ on the interval $0 \leq \tau \leq 1$:

$$\begin{aligned} \frac{d}{d\tau} \left[\frac{\partial \mathbf{x}(\tau)}{\partial p_i} \right] &= F(\tau) \frac{\partial \mathbf{x}(\tau)}{\partial p_i} + \mathbf{u}(\tau) \frac{\partial P'_f}{\partial p_i} + \mathbf{u}_i(\tau) \\ 0 &= G_0 \frac{\partial \mathbf{x}(0)}{\partial p_i} + \mathbf{w}_0 \frac{\partial P'_f}{\partial p_i} + \mathbf{w}_{0i} \\ 0 &= G_f \frac{\partial \mathbf{x}(1)}{\partial p_i} + \mathbf{w}_f \frac{\partial P'_f}{\partial p_i} \end{aligned} \quad (37)$$

where

$$\begin{aligned} \mathbf{u}_i(\tau) &= \left. \frac{\partial \mathbf{f}}{\partial p_i} \right|_{[\mathbf{x}(\tau), P'_f, \mathbf{p}]} \\ G_0 &= \left. \frac{\partial \mathbf{g}_0}{\partial \mathbf{x}} \right|_{[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0]} \\ \mathbf{w}_0 &= \left. \frac{\partial \mathbf{g}_0}{\partial P'_f} \right|_{[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0]} \\ \mathbf{w}_{0i} &= \left. \frac{\partial \mathbf{g}_0}{\partial p_i} \right|_{[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0]} \\ G_f &= \left. \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}} \right|_{[\mathbf{x}(1), P'_f, \vec{\mathbf{r}}_f]} \\ \mathbf{w}_f &= \left. \frac{\partial \mathbf{g}_f}{\partial P'_f} \right|_{[\mathbf{x}(1), P'_f, \vec{\mathbf{r}}_f]} \end{aligned} \quad (38)$$

The linear TPBVP in Eq. (37) can be solved using state transition matrix methods that are fairly standard. Such methods can be used to develop a formula for $\partial \mathbf{x}(1) / \partial p_i$ that involves a non-homogeneous term and terms that are linear in $\partial \mathbf{x}(0) / \partial p_i$ and $\partial P'_f / \partial p_i$. This expression

can be substituted into the third line of Eq. (37). The resulting equation, when combined with the second line of Eq. (37), yields a well-defined linear system of equations for the unknown quantities $\partial \mathbf{x}(0)/\partial p_i$ and $\partial P'_f/\partial p_i$. The solution for these quantities and the ODE in the first line of Eq. (37) can be used along with numerical integration to determine $\partial \mathbf{x}(\tau)/\partial p_i$ at another other value of τ .

B. Rationale for Using Adjoint Calculation of Partial Derivatives

The calculations of the preceding subsection are straightforward to implement. Note, however, that a new linear TPBVP would have to be solved for each element of the parameter vector \mathbf{p} . The only change from element to element would be the non-homogeneous terms in the TPBVP. Nevertheless, the computational cost of re-solving many TPBVPs could be prohibitive.

An alternate approach to partial derivative calculation uses an adjoint. Before developing such an approach for the TPBVP in Eq. (37), it is useful to review the reason why an adjoint calculation can reduce the number of computations.

Consider a simple vector space example of using an adjoint. Suppose that the $n_x \times 1$ vector x is an implicit function of the $n_p \times 1$ vector p through the $n_x \times 1$ vector function f :

$$0 = f(x, p) \quad (39)$$

and suppose that the $n_y \times 1$ vector y is an explicit function of x :

$$y = h(x) \quad (40)$$

The partial derivative of y with respect to p can be calculated by first taking the partial derivative of Eq. (39) with respect to p and using the result to solve for $\partial x/\partial p$. Afterwards, the partial derivative of Eq. (40) can be taken with respect to p , and the formula for $\partial x/\partial p$ can be substituted into the result in order to yield:

$$\begin{aligned} \frac{\partial y}{\partial p} &= -\frac{\partial h}{\partial x} \left(\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial p} \end{bmatrix}^{-1} \frac{\partial f}{\partial p} \right) \\ &= -\left(\frac{\partial h}{\partial x} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial p} \end{bmatrix}^{-1} \right) \frac{\partial f}{\partial p} \end{aligned} \quad (41)$$

The only difference between the right-hand sides on the two lines of this equation is the order of their matrix-matrix multiplications. The upper line is the direct-method calculation of $\partial y/\partial p$, and the lower line is the adjoint calculation. The transpose of the term in parentheses on the second line is the calculation's adjoint. These two versions of the same matrix-matrix multiplications have

different operation counts. The direct method on the first line employs $n_x^2 n_p + n_y n_x n_p$ scalar multiplications. The adjoint method on the second line employs $n_x^2 n_y + n_y n_x n_p$ multiplications. If $n_y < n_p$, then the adjoint method is more efficient. If $n_y \ll n_p$ and n_x is large, then the computational burden of the adjoint method may be orders of magnitude lower than that of the direct method.

Similar considerations carry over to the sensitivities of the solution of this paper's TPBVP. The corresponding potential for computational cost savings constitutes the rationale for developing an adjoint method to compute the TPBVP's partial derivatives.

C. Output Functions to be Differentiated

The preceding discussion justifies the use of an adjoint method if the number of problem outputs is much lower than the number of inputs. Furthermore, the adjoint itself is calculated using partial derivatives of the outputs. Therefore, it is necessary to define the TPBVP outputs whose partial derivatives are to be computed.

The problem outputs of interest are the range-equivalent total group delay P'_f , the range-equivalent terminal carrier phase $P(1)$, the initial wave vector $\vec{k}(0)$, and the terminal wave vector $\vec{k}(1)$. Output of the terminal carrier phase implies that the adjoint calculations must work with a state-space model for the 7-element state vector that is defined in Eq. (5). Therefore, all developments in this section presume the use of this state vector and use of the corresponding 7-element $\mathbf{f}(\mathbf{x}, P'_f, \mathbf{p})$ dynamics model function. These developments also assume that the ray-tracing TPBVP solution values for $\mathbf{x}(\tau)$ and P'_f have been computed.

Use of the 7-element state vector necessitates a modification of the generalized initial condition function $g_0[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0]$ in the first line of Eq. (19). It must be augmented by appending the expression $[0, 0, 0, 0, 0, 0, 1]\mathbf{x}(0)$, which changes it from a 4-element vector function into a 5-element function. This added initial condition specifies that the initial carrier phase is 0 so that $P(1)$ gives the range-equivalent change in the carrier phase over the ray path.

Given the 7-element form of $\mathbf{x}(\tau)$, all desired outputs of the TPBVP can be written in terms of the initial state $\mathbf{x}(0)$, the terminal state $\mathbf{x}(1)$, and the terminal range-equivalent group delay P'_f . The following two general formulas cover all possible desired outputs:

$$\begin{aligned} \mathbf{y}_0 &= \mathbf{h}_0[\mathbf{x}(0), P'_f] \\ \mathbf{y}_f &= \mathbf{h}_f[\mathbf{x}(1), P'_f] \end{aligned} \quad (42)$$

Adjoint partial derivative calculations will be developed for these two generalized output vectors.

In preparation for developing the adjoint method, it is helpful to define the following partial derivatives of the vector functions \mathbf{h}_0 and \mathbf{h}_f :

$$\begin{aligned} H_0 &= \left. \frac{\partial h_0}{\partial \mathbf{x}} \right|_{[\mathbf{x}(0), P'_f]} \\ z_0 &= \left. \frac{\partial h_0}{\partial P'_f} \right|_{[\mathbf{x}(0), P'_f]} \\ H_f &= \left. \frac{\partial h_f}{\partial \mathbf{x}} \right|_{[\mathbf{x}(1), P'_f]} \\ z_f &= \left. \frac{\partial h_f}{\partial P'_f} \right|_{[\mathbf{x}(1), P'_f]} \end{aligned} \quad (43)$$

Note that the Jacobian matrices H_0 and H_f must not be confused with the scalar Hamiltonian functions H_1 and H_2 that are defined in Section II. Also, the generalized output vector functions \mathbf{h}_0 and \mathbf{h}_f must not be confused with the WGS-84 altitude function h_{alt} that is defined and used in Subsection III-C.

D. Adjoint Calculation of Partial Derivatives of Outputs with Respect to Ionosphere Parameters

This subsection presents the adjoint method for calculating the partial derivative of the initial output \mathbf{y}_0 and the terminal output \mathbf{y}_f with respect to p_i , the i^{th} element of the ionosphere parameter vector \mathbf{p} . The formulas presented in this section have been derived by first developing a direct-method solution using state transition matrix methods, as discussed briefly in Subsection IV-A. The resulting solution was then manipulated to derive its adjoint equivalent. The necessary derivations are lengthy and have been omitted for the sake of brevity.

The adjoint solution starts by solving a pair of linear matrix final-value problem:

$$\begin{aligned} \frac{d\Lambda_g(\tau)}{d\tau} &= -F^T(\tau)\Lambda_g(\tau) \\ \Lambda_g(1) &= G_f^T \\ \frac{d\Lambda_h(\tau)}{d\tau} &= -F^T(\tau)\Lambda_h(\tau) \\ \Lambda_h(1) &= H_f^T \end{aligned} \quad (44)$$

These two solutions are used to compute three vectors. The first two are terminal group delay adjoint vectors, and the third projects the parameter p_i 's differential equation

right-hand-side sensitivities onto the terminal boundary condition:

$$\begin{aligned} \boldsymbol{\lambda}_g &= \mathbf{w}_f + \int_0^1 \Lambda_g^T(\tau)\mathbf{u}(\tau)d\tau \\ \boldsymbol{\lambda}_h &= \mathbf{z}_f + \int_0^1 \Lambda_h^T(\tau)\mathbf{u}(\tau)d\tau \\ \boldsymbol{\mu}_{gi} &= \int_0^1 \Lambda_g^T(\tau)\mathbf{u}_i(\tau)d\tau \\ \boldsymbol{\mu}_{hi} &= \int_0^1 \Lambda_h^T(\tau)\mathbf{u}_i(\tau)d\tau \end{aligned} \quad (45)$$

Note: the adjoint method vectors $\boldsymbol{\lambda}_g$ and $\boldsymbol{\lambda}_h$ must not be confused with the eigenvalue λ that is used to calculate the Hamiltonian H_2 in Subsection II-B.

Next the adjoint matrices of the initial and terminal boundary conditions are computed for the initial and terminal outputs:

$$\begin{bmatrix} \Lambda_{0a} & \Lambda_{0b} \\ \Lambda_{fa} & \Lambda_{fb} \end{bmatrix} = \begin{bmatrix} G_0^T & \Lambda_g(0) \\ \mathbf{w}_0^T & \boldsymbol{\lambda}_g^T \end{bmatrix}^{-1} \begin{bmatrix} H_0^T & \Lambda_h(0) \\ \mathbf{z}_0^T & \boldsymbol{\lambda}_h^T \end{bmatrix} \quad (46)$$

Finally, the initial and terminal output sensitivities are computed:

$$\begin{bmatrix} \frac{\partial \mathbf{y}_0}{\partial p_i} \\ \frac{\partial \mathbf{y}_f}{\partial p_i} \end{bmatrix} = - \begin{bmatrix} \Lambda_{0a}^T & \Lambda_{fa}^T \\ \Lambda_{0b}^T & \Lambda_{fb}^T \end{bmatrix} \begin{bmatrix} \mathbf{w}_{0i} \\ \boldsymbol{\mu}_{gi} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\mu}_{hi} \end{bmatrix} \quad (47)$$

This is a very efficient computation because the only calculations that need to be re-done for a different parameter index i are those in the fourth line of Eq. (38), those in the last two lines of Eq. (45), and those in Eq. (47).

E. Adjoint Calculation of Partial Derivatives of Outputs with Respect to Initial and Terminal Positions

The calculation of output partial derivatives with respect to the initial and final ray-path locations $\vec{\mathbf{r}}_0$ and $\vec{\mathbf{r}}_f$ re-uses many of the quantities that are computed as part of the p_i partial derivative calculation. The only new quantities are the partial derivatives of the initial and terminal boundary conditions with respect to the initial and terminal positions:

$$\begin{aligned} G_{0r} &= \left. \frac{\partial \mathbf{g}_0}{\partial \vec{\mathbf{r}}} \right|_{[\mathbf{x}(0), P'_f, \mathbf{p}, \vec{\mathbf{r}}_0]} \\ G_{fr} &= \left. \frac{\partial \mathbf{g}_f}{\partial \vec{\mathbf{r}}} \right|_{[\mathbf{x}(1), P'_f, \vec{\mathbf{r}}_f]} \end{aligned} \quad (48)$$

The desired output partial derivatives are then

$$\begin{bmatrix} \frac{\partial \mathbf{y}_0}{\partial \vec{\mathbf{r}}_0} & \frac{\partial \mathbf{y}_0}{\partial \vec{\mathbf{r}}_f} \\ \frac{\partial \mathbf{y}_f}{\partial \vec{\mathbf{r}}_0} & \frac{\partial \mathbf{y}_f}{\partial \vec{\mathbf{r}}_f} \end{bmatrix} = - \begin{bmatrix} \Lambda_{0a}^T & \Lambda_{fa}^T \\ \Lambda_{0b}^T & \Lambda_{fb}^T \end{bmatrix} \begin{bmatrix} G_{0r} & 0 \\ 0 & G_{fr} \end{bmatrix} \quad (49)$$

F. Proper Numerical Integration in Adjoint Calculations

Care must be taken in the numerical integration of the matrix differential equations in Eq. (44) and the integrals in Eq. (45). One might be tempted to calculate them using the Dormand-Prince 4/5 method or a similar method.

The matrix final value problems in Eq. (44) can be integrated backward in time from the terminal condition at $\tau = 1$. The integrals in Eq. (45) can be evaluated by simultaneously integrating the following vector final value problems:

$$\begin{aligned} \frac{d\tilde{\boldsymbol{\lambda}}_g(\tau)}{d\tau} &= -\Lambda_g^T(\tau)\mathbf{u}(\tau) \\ \tilde{\boldsymbol{\lambda}}_g(1) &= \mathbf{w}_f \\ \frac{d\tilde{\boldsymbol{\lambda}}_h(\tau)}{d\tau} &= -\Lambda_h^T(\tau)\mathbf{u}(\tau) \\ \tilde{\boldsymbol{\lambda}}_h(1) &= \mathbf{z}_f \\ \frac{d\tilde{\boldsymbol{\mu}}_{gi}(\tau)}{d\tau} &= -\Lambda_g^T(\tau)\mathbf{u}_i(\tau) \\ \tilde{\boldsymbol{\mu}}_{hi}(1) &= 0 \\ \frac{d\tilde{\boldsymbol{\mu}}_{hi}(\tau)}{d\tau} &= -\Lambda_h^T(\tau)\mathbf{u}_i(\tau) \\ \tilde{\boldsymbol{\mu}}_{hi}(1) &= 0 \end{aligned} \quad (50)$$

backward in time from $\tau = 1$ and setting $\boldsymbol{\lambda}_g = \tilde{\boldsymbol{\lambda}}_g(0)$, $\boldsymbol{\lambda}_h = \tilde{\boldsymbol{\lambda}}_h(0)$, $\boldsymbol{\mu}_{gi} = \tilde{\boldsymbol{\mu}}_{gi}(0)$, and $\boldsymbol{\mu}_{hi} = \tilde{\boldsymbol{\mu}}_{hi}(0)$.

The backward numerical integrations in the first line of Eq. (44) and in the first and fifth lines of Eq. (50) must be carried out together as though a single large “state” were being integrated that consisted of the columns of $\Lambda_g(\tau)$, the vector $\tilde{\boldsymbol{\lambda}}_g$, and the vector $\tilde{\boldsymbol{\mu}}_{gi}$. If multiple $\tilde{\boldsymbol{\mu}}_{gi}$ are being evaluated for multiple parameter vector element indices i , then all of them must be included in the large “state”. Similarly, the backward numerical integrations in the third line of Eq. (44) and in the third and seventh lines of Eq. (50) must be carried out together as though a single large “state” were being integrated that consisted of the columns of $\Lambda_h(\tau)$, the vector $\tilde{\boldsymbol{\lambda}}_h$, and all $\tilde{\boldsymbol{\mu}}_{hi}$ vectors.

These backward integrations must not be carried out using the same Runge-Kutta numerical integration method as has been used to perform forward integration of Eq. (3). Instead, a special adjoint of the particular Runge-Kutta method must be used. The adjoint of each Runge-Kutta

step’s sequence of forward integration formulas is also a sequential calculation. It performs reverse-time numerical integration using an altered Runge-Kutta computation recipe that is based on the forward method’s same Runge-Kutta tableau. The calculations of the adjoint Runge-Kutta method are given in Eqs. (15) and (16) of [11].

The backward numerical integration of Eqs. (44) and (50) requires various values of $F(\tau)$, $\mathbf{u}(\tau)$, and $\mathbf{u}_i(\tau)$ during each integration step. The values that must be used are computed during the forward numerical integration of Eq. (3) [11]. They can be stored for later use in the backward numerical integration of Eqs. (44) and (50).

G. Inefficiency of Finite Difference Sensitivity Calculations

One might be tempted to avert the difficulty of implementing the adjoint partial derivative calculations of this section. An alternative would be to approximate the sensitivities by using finite differences. The use of finite differences in the present case is not recommended. There are two reasons for this recommendation. Both stem from the fact that each finite difference calculation will require iterative re-resolution of the TPBVP.

Re-resolution of the TPBVP is costly in terms of computation time, and the expensive TPBVP solution would need to be carried out on the order of n_p times. Recall that n_p is the number of elements in the ionosphere parameter vector \mathbf{p} . Each TPBVP solution will require at least one expensive Newton iteration. The large number of TPBVP solutions will result in a computation time that scales like the inefficient direct sensitivities calculation method discussed in Subsection IV-B. Therefore, finite-difference sensitivity calculation could prove very expensive computationally.

A second reason to avoid finite-difference calculations is the impact of possible residual errors in TPBVP solutions. Suppose that the Newton method has not fully converged for a reported TPBVP solution that is being used in a finite-difference calculation. Then the small TPBVP solution error will contribute directly to an error in the finite-differenced partial derivative approximation. If the finite-difference increment is very small, then this error can be large. If the finite-difference increments are large enough to make such errors negligible, then an increased amount of truncation error might adversely impact the finite-difference sensitivity approximations.

V. EXAMPLE RAY-TRACING SOLUTIONS

The ray-tracing TPBVP solution algorithms of Section III have been encoded in MATLAB and tested on example problems. These problems include O-mode and X-mode propagation. Some consider the reflection case that models an ionosonde with transmitter and receiver collocated, and others consider the point-to-point case. The X-mode cases include situations where the transmission frequency is higher than the electron gyro frequency ($Y < 1$) and situations where it is lower than the gyro frequency ($Y > 1$). Two different transmission frequencies have been used, 1.09 MHz and 11.23 MHz.

Terminal range-equivalent group delays P_f' vary between 294.5 km and 381.7 km for the ionosonde reflection cases and between 589.0 km and 903.0 km for the point-to-point cases. One expects the reflection cases to be shorter because they only trace the path up to the reflection point. The total range-equivalent group delay between ionosonde transmission and reception would be twice this number. Note that P_f' for the reflection cases is equivalent to ionosonde virtual height.

Newton's method successfully converged in 3-5 iterations for all cases considered. This count includes iterations of the first TPBVP solver that used a coarse Dormand-Prince solution accuracy and iterations of the second TPBVP solver that used a finer set of integration intervals to enforce a finer numerical integration accuracy. Typically only a single additional Newton iteration is required when using the finer numerical integration grid points because the TPBVP solutions for \hat{s} and P_f' on the coarse grid already have good accuracy.

The first-guess generation procedure's third stage always selected candidate \hat{s}_a as the best guess of the initial wave vector direction. Recall from Eq. (33) that this is the guess generated by the second stage of the first-guess procedure. These guesses always resulted in rapid convergence of the Newton/shooting algorithm. Therefore, the process of trying all 5 first guesses from Eq. (33) may be unnecessary for ensuring robust TPBVP solution convergence.

Example TPBVP solutions are presented in Figs. 2, 3, and 4. Figures 2 and 3 show, respectively, the 3D ray path $\vec{r}(\tau)$ and the corresponding non-dimensionalized wave vector time history $\vec{k}(\tau)/k_0$ vs. τ for an 11.23 MHz O-mode signal. Both paths show smooth ray bending in the ionosphere in the range $0.4 \leq \tau \leq 0.6$. Although not directly plotted, one can infer from the results in Fig. 3 that the minimum value of $\|\vec{k}(\tau)/k_0\|$ is 0.67. Thus, this ray path does not penetrate high enough into the ionosphere to have the possibility of encountering a spitzze singularity.

Figure 4 shows the upper portion of the 3D ray path for a

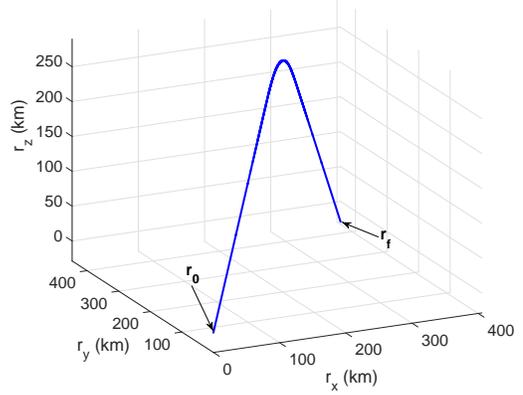


Fig. 2: Example point-to-point ray path solution for an 11.23 MHz O-mode signal.

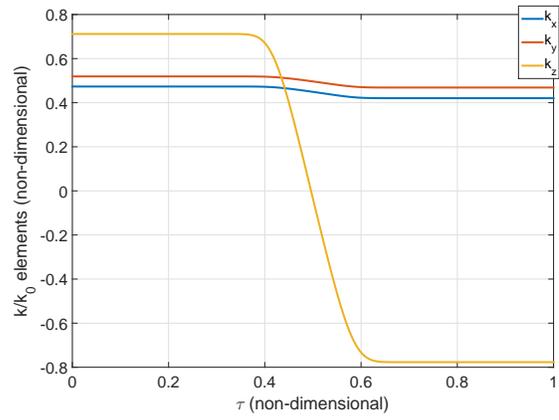


Fig. 3: Non-dimensionalized wave vector solution time history for the point-to-point ray path of an 11.23 MHz O-mode signal.

1.09 MHz X-mode signal in the situation of $Y > 1$. This upper portion includes an unusual curving portion near its very top that is characteristic of a spitzze singularity. Thus, the new ray-tracing method can handle a spitzze.

VI. SUMMARY AND CONCLUSIONS

This paper has developed and tested a new method of solving for HF radio-wave ray paths through the ionosphere. It has also developed a method for calculating the partial derivative sensitivities of ray-path observables with respect to ray-path input parameters. For example, the partial derivative of the range-equivalent group delay can be calculated with respect to the height of a Chapman profile's peak electron density that applies at a node of a latitude/longitude spline.

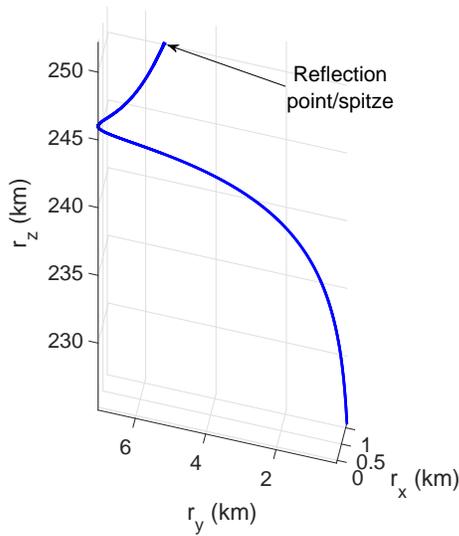


Fig. 4: Upper segment of an ionosonde reflection ray path solution for an 1.09 MHz X-mode signal with $Y > 1$.

The ray-tracing algorithm solves a nonlinear two-point boundary value problem. It uses a Hamiltonian system of differential equations that model the Appleton-Hartree index of refractive and its effect on radio wave propagation. The solver combines the shooting algorithm with Newton's method in order to achieve rapid convergence. The implementation of Newton's method includes the solution of auxiliary linear two-point boundary value problems in order to determine state vector partial derivative time histories. These time histories enable the computation of Newton increments to the initial wave vector's 2 aiming angles and to the terminal range-equivalent group delay. An *ad hoc* procedure for generating first guesses augments the Newton algorithm. It provides good first guesses of the aiming angles and the group delay so that Newton's method typically converges in 4 iterations or fewer.

The partial derivative sensitivity algorithm is based on the adjoint method of differential equation theory. It avoids the need to solve many state-vector sensitivity two-point boundary value problems by solving a set of adjoint differential equations. These equations must be integrated backward in time using a special adjoint numerical integration method.

The ray-tracing algorithm has been tested on representative problems. It has successfully solved ionosonde reflection problems and point-to-point problems between known transmitter and receiver locations. It works for both O-mode and X-mode radio wave propagation, and it can trace a ray path through a spitz reflection singularity without

any difficulty.

REFERENCES

- [1] K. Chiang and M. Psiaki, "GPS and Ionosonde Data Fusion for Local Ionospheric Parameterization," in *Proc. ION GNSS+ 2014*. Tampa, FL: Institute of Navigation, Sept. 2014, pp. 1163–1172.
- [2] Y. Baumgarten and M. Psiaki, "Navigation Using High-Frequency Ground Beacons and Ionosphere Model Corrections," in *Proc. ION GNSS+ 2015*. Tampa, FL: Institute of Navigation, Sept. 2015, pp. 1425–1435.
- [3] D. Hysell, M. Milla, and J. Vierinen, "A Multistatic HF Beacon Network for Ionospheric Specification in the Peruvian Sector," *Radio Science*, vol. 51, no. 5, pp. 392–401, May 2016.
- [4] R. Jones and J. Stephenson, "A Versatile Three-Dimensional Ray Tracing Computer Program for Radio Waves in the Ionosphere," U.S. Dept. of Commerce, OT Report 75-76, Oct. 1975.
- [5] K. Yeh and C. Liu, *Theory of Ionospheric Waves*. New York: Academic Press, 1972.
- [6] Anon. (2014, Dec.) International geomagnetic reference field. NOAA. Boulder, CO. [Online]. Available: <http://www.ngdc.noaa.gov/AGA/vmod/igrf.html>
- [7] P. Gill, W. Murray, and M. Wright, *Practical Optimization*. New York: Academic Press, 1981.
- [8] J. Dormand and P. Prince, "A family of Embedded Runge-Kutta Formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [9] M. Psiaki, G. Bust, and C. Mitchell, "Nonlinear Estimation to Assimilate GPS TEC Data into a Regional Ionosphere Model," in *Proc. ION GNSS+ 2015*. Tampa, FL: Institute of Navigation, Sept. 2015, pp. 3497–3510.
- [10] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *GPS: Theory and Practice, 5th Edition*. New York: Springer-Verlag, 2001.
- [11] W. Hager, "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, vol. 87, no. 2, pp. 247–282, 2000.