infraGA/GeoAc

Numerical tools to model infrasonic propagation in the limit of geometric acoustics

Author/Maintainer: Philip S. Blom (pblom@lanl.gov)

Contents

1 Introduction and Installation

InfraGA/GeoAc is a numerical package written in C++, which solves the equations governing acoustic propagation through the atmosphere in the geometric limit using the geometric acoustic methods in the GeoAc library via a classical fourth-order Runge Kutta (RK4) algorithm. The toolkit contains multiple instances of said equation system and is able to model propagation in an azimuthal plane using the effective motionless medium approximation as well as in three dimensions using an inhomogeneous moving background medium. The three dimensional propagation scheme include methods to model propagation in a Cartesian coordinate system as well as a curvilinear coordinate system which incorporates the curvature of the earth as either spherical or elliptical (as defined by the WGS84 ellipsoid).

In addition to the geometric propagation paths which are straightforward to compute, the GeoAc library uses auxiliary parameters as discussed in *Impulse propagation in the nocturnal boundary layer - analysis of the geometric component* [Blom & Waxler; J. Acoust. Soc. Am., 2012, Vol.131(5), pp.3680–90] to calculate the Jacobian determinant and obtain a frequency independent amplitude coefficient describing the attenuation due to geometric spreading. The auxiliary parameters used to compute the amplitude coefficient have been found to provide an efficient means to identify eigenrays (propagation paths connecting a source and receiver at specific locations) as detailed in *Modeling and observations of an elevated, moving infrasonic source: eigenray methods* [Blom & Waxler; J. Acoust. Soc. Am., 2016, In Review]. Eigenray information provides a means to predict the characteristics of the various acoustic phases observed at a given receiver location for a known source and propagation medium.

For propagation through a horizontally varying medium, a multivariate interpolation scheme has been developed which uses a modified Keys bicubic interpolation algorithm as described in *Cubic Convolution Interpolation for Digital Image Processing* [IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing 29 (6): pp.1153-60]. In this case, the interpolation algorithm has been configured so that both first and second order derivatives of the medium are continuous between grid squares. This condition is required to generate smooth and continuous solutions for the auxiliary parameters used in computation the geometric spreading.

Topographical features can be incorporated into the propagation modeling, though the required interpolation precision significantly increases computation time. Methods currently allow for topographical effects in a plane (using the effective sound speed approximation for the atmosphere) or for a full two dimensional topographical structure which produces significant non-planar propagation in the case of topographical gradients perpendicular to the azimuth of propagation. Note that the topographical effects modeled in the geometric limit neglect diffraction and scattering.

Installation:

- Ensure that dependencies are installed: `fftw`, `OpenMPI` (optional).
- Open a terminal, `cd` into the directory containing the infraGA/GeoAc materials
- To create the standard methods within the local directory structure, run `make`.
- If you have `OpenMPI`, you can build the multi-threaded version of infraGA by running `make accel`.

The rest of this manual is divided up into the following sections: explanation of the usage of the individual executables are detailed in the Sect. 2, clarification of the amplitude computation and range dependent profile interpolation are summarized in Sect. 3, details of additional parameters available for all executables to customize propagation are given in Sect. 4. An overview of the changes made during each revision of the package are included in Sect. 5.

2 Using infraGA

All of the methods in InfraGA have usage output that is displayed if no arguments are provided to the call. These usage summaries have example usage calls that can be helpful in understanding how to run the code. Additional details for each method are provided below.

2.1 2D Stratified Cartesian Propagation

Description

The methods in `infraga-2d` compute ray paths in an azimuthal plane using the effective sound speed approximation. These methods are useful for comparing predictions with other propagation schemes (normal mode, parabolic equation, etc.) that require the effective sound speed.

Usage

```
infraga-2d  [option]  profile.met  [parameters]
```

- Only one option can be applied for a given run. Possible options are:

`-prop`	Generate ray paths at a fixed azimuth using multiple inclination angles.
`-wnl_wvfrm`	Compute the weakly non-linear waveform evolution along a single ray path.

- The `profile.met` file is expected to contain columns describing the atmosphere in the format:

$$z\,[\mathrm{km}] \vdots T(z)\,[\mathrm{K}] \vdots u(z) \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots v(z) \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots \rho_0(z) \left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \vdots p_0(z)\,[\mathrm{mbar}]$$

- Parameters are set using the format `parameter_name=value`, for example: `incl_step=1.0`. Possible parameters for each option are included below.

Parameters for propagation option		
Parameter	Description	Default Value
`incl_min`	Minimum inclination angle (0^o = horizontal)	0.5^o
`incl_max`	Maximum inclination angle (0^o = horizontal)	45.0^o
`incl_step`	Inclination step size	0.5^o
`azimuth`	Azimuth angle (North = 0^o, increases clockwise)	-90.0^o
`bounces`	Maximum # of bounces to compute (integer ≥ 0)	2
`src_alt`	Altitude of the source (rel. sea level)	0.0 km

Parameters for weakly non-linear waveform option		
Parameter	Description	Default Value
`inclination`	Initial inclination angle of the ray	10^o
`azimuth`	Initial azimuth angle of the ray	90^o
`bounces`	# of bounces to compute (integer ≥ 0)	0
`src_alt`	Elevation of the source	0.0 km
...	See Section 4.8	...

Output

`atmo.dat`
Contains the interpolate atmospheric profile. Columns are:

$$z\,[\mathrm{km}] \vdots c(z) \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots u(z) \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots v(z) \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots \rho_0(z) \left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \vdots c_{\mathrm{eff}}\,(z, \varphi_0) \left[\frac{\mathrm{m}}{\mathrm{s}}\right]$$

`file_name.raypaths.dat`
Contains the ray path information. Columns are:

$$r[\text{km}] \;\vdots\; z[\text{km}] \;\vdots\; A_{\text{geo.}}[\text{dB}] \;\vdots\; A_{\text{atmo.}}[\text{dB}] \;\vdots\; t[\text{sec}]$$

`file_name.arrivals.dat`
Contains the arrival information for each ground intercept

$$\vartheta[\text{deg}] \vdots \varphi[\text{deg}] \vdots n_{\text{bnc}} \vdots r_0[\text{km}] \vdots t[\text{sec}] \vdots \nu[\text{km/s}] \vdots z_{\text{max}}[\text{km}] \vdots \vartheta_{\text{inc.}}[\text{deg}] \vdots \varphi_{\text{back}}[\text{deg}] \vdots A_{\text{geo.}}[\text{dB}] \vdots A_{\text{atmo.}}[\text{dB}]$$

2.2 3D Stratified Cartesian Propagation

Description

The methods in `infraga-3d` compute ray paths in a three dimensional inhomogeneous moving medium. In this case, the atmosphere is assumed to vary only with altitude and be specified by a single file. Because of this horizontal symmetry, the source is assumed to be at the origin though its elevation above the ground can be modified.

Usage

```
infraga-3d [option] profile.met [parameters]
```

- Only one option can be applied for a given run. Possible options are:

`-prop`	Generate ray paths at a multiple azimuths and multiple inclination angles.
`-eig_search`	Search for all eigenray rays within a chosen inclination range connecting a source at $(0, 0, z_{\text{src}})$ to a receiver at $(x_{\text{rcvr}}, y_{\text{rcvr}}, z_{\text{grnd}})$.
`-eig_direct`	Search for a specific eigenray near estimated inclination and azimuth angles connecting a source at $(0, 0, z_{\text{src}})$ to a receiver at $(x_{\text{rcvr}}, y_{\text{rcvr}}, z_{\text{grnd}})$.
`-wnl_wvfrm`	Compute the weakly non-linear waveform evolution along a single ray path.

- The `profile.met` file is expected to contain columns describing the atmosphere in the format:

$$z\,[\text{km}] \;\vdots\; T(z)\,[\text{K}] \;\vdots\; u(z)\left[\frac{\text{m}}{\text{s}}\right] \;\vdots\; v(z)\left[\frac{\text{m}}{\text{s}}\right] \;\vdots\; \rho_0(z)\left[\frac{\text{g}}{\text{cm}^3}\right] \;\vdots\; p_0(z)\,[\text{mbar}]$$

- Parameters are set using the format `parameter_name=value`, for example: `incl_step=1.0`. Possible parameters for each option are included below.

Parameters for propagation option		
Parameter	Description	Default Value
`incl_min`	Minimum inclination angle (0^o = horizontal)	0.5^o
`incl_max`	Maximum inclination angle (0^o = horizontal)	45.0^o
`incl_step`	Inclination step size	0.5^o
`inclination`	Set a *single* inclination (`incl_max` = `incl_min`)	–
`az_min`	Minimum azimuth angle (North = 0^o, increases clockwise)	-90.0^o
`az_max`	Maximum azimuth angle (North = 0^o, increases clockwise)	-90.0^o
`az_step`	Azimuth step size	1.0^o
`azimuth`	Set a *single* azimuth (`az_max` = `az_min`)	–
`bounces`	Maximum # of bounces to compute (integer ≥ 0)	2
`src_x`	East/West offset of the source	0.0 km
`src_y`	North/South offset of the source	0.0 km
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`write_atmo`	Boolean to write the atmosphere specifications	false
`write_rays`	Boolean to write the ray paths	true

Parameters for eigenray search option		
Parameter	Description	Default Value
`incl_min`	Minimum inclination angle (0^o = horizontal)	0.5^o
`incl_max`	Maximum inclination angle (0^o = horizontal)	45.0^o
`bnc_min`	Minimum # of bounces to consider (integer)	0
`bnc_max`	Maximum # of bounces to consider (integer)	0
`bounces`	Exact # of bounces (`bnc_min` = `bnc_max`)	0
`src_x`	East/West offset of the source	0.0 km
`src_y`	North/South offset of the source	0.0 km
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_x`	East/West offset of receiver	−250.0 km
`rcvr_y`	North/South offset of receiver	0.0 km
`verbose`	Boolean to output details of the search	false
`iterations`	Maximum number of iterations in eigenray search	25
`damping`	Damping factor in Levenberg-Marquardt solver	1.0e-3
`tolerance`	Accepted distance from receiver location	0.1 km
`incl_step_min`	Smallest step size in search algorithm	1.0e-3^o
`incl_step_max`	Largest step size in search algorithm	0.1^o

Parameters for eigenray direct option		
Parameter	Description	Default Value
`incl_est`	Estimated inclination angle (0^o = horizontal)	15.0^o
`az_est`	Estimated azimuth angle (North = 0^o, increases clockwise)	atan2 (`y_rcvr`, `x_rcvr`)
`bounces`	Exact # of bounces to consider (integer ≥ 0)	0
`src_x`	East/West offset of the source	0.0 km
`src_y`	North/South offset of the source	0.0 km
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_x`	East/West offset of receiver	−250.0 km
`rcvr_y`	North/South offset of receiver	0.0 km
`verbose`	Boolean to output details of the search	false
`iterations`	Maximum number of iterations in eigenray search	25
`damping`	Damping factor in Levenberg-Marquardt solver	1.0e-3
`tolerance`	Accepted distance from receiver location	0.1 km

Parameters for weakly non-linear waveform option		
Parameter	Description	Default Value
`inclination`	Initial inclination angle of the ray	10^o
`azimuth`	Initial azimuth angle of the ray	90^o
`bounces`	# of bounces to compute (integer ≥ 0)	0
`src_x`	East/West offset of the source	0.0 km
`src_y`	North/South offset of the source	0.0 km
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
...	See Section 4.8	...

Output

`atmo.dat`
Contains the interpolate atmospheric profile. Columns are:

$$z\,[\mathrm{km}] \;\vdots\; c(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \;\vdots\; u(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \;\vdots\; v(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \;\vdots\; \rho_0(z)\left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \;\vdots\; c_{\mathrm{eff}}\left(z,\varphi_0\right)\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$$

`file_name.raypaths.dat`
`file_name.eigenray-#.dat`
Contains the ray path information. Columns are:

$$x[\text{km}] \;\vdots\; y[\text{km}] \;\vdots\; z[\text{km}] \;\vdots\; A_{\text{geo.}}[\text{dB}] \;\vdots\; A_{\text{atmo.}}[\text{dB}] \;\vdots\; t[\text{sec}]$$

`file_name.arrivals.dat`
Contains the arrival information for each ground intercept for the `-prop` option:

$$\vartheta[\text{deg}] \;\vdots\; \varphi[\text{deg}] \;\vdots\; n_{\text{bnc}} \;\vdots\; x_0[\text{km}] \;\vdots\; y_0[\text{km}] \;\vdots\; t[\text{sec}] \;\vdots\; \nu[\text{km/s}] \;\vdots\; z_{\max}[\text{km}] \;\vdots\; \vartheta_{\text{inc.}}[\text{deg}] \;\vdots\; \varphi_{\text{back}}[\text{deg}] \;\vdots\; A_{\text{geo.}}[\text{dB}] \;\vdots\; A_{\text{atmo.}}[\text{dB}]$$

Contains the run summary and arrival information for each eigenray for the `-eig_search` option:

```
Eigenray-#.  [ ] bounce(s).
      vartheta, phi = [ ].
      Travel Time = [ ].
      Celerity = [ ].
      Amplitude (geometric) = [ ].
      Atmospheric attenuation =  [ ].
      Arrival inclination =  [ ].
      Azimuth to source = [ ].
      Back azimuth of arrival = [ ].
      Azimuth deviation = [ ] .
```

2.3 3D Stratified Spherical Propagation

Description

The methods in `infraga-sph` compute ray paths in a three dimensional inhomogeneous moving medium using spherical coordinates. The earth is modeled as a sphere with a fixed radius of 6,370 kilometers. The atmosphere is assumed to be stratified and specified by a single file. The latitude and longitude of the source can be specified to produce results for specific geographic locations without requiring coordinate shifts.

Usage

```
infraga-sph  [option]  profile.met  [parameters]
```

- Only one option can be applied for a given run. Possible options are:

`-prop`	Generate ray paths at a multiple azimuths and multiple inclination angles.
`-eig_search`	Search for all eigenray rays within a chosen inclination range connecting a source at $(\text{lat}_{\text{src}}, \text{lon}_{\text{src}}, z_{\text{src}})$ to a receiver at $(\text{lat}_{\text{rcvr}}, \text{lon}_{\text{rcvr}}, z_{\text{grnd}})$.
`-eig_direct`	Search for a specific eigenray near estimated inclination and azimuth angles connecting a source at $(\text{lat}_{\text{src}}, \text{lon}_{\text{src}}, z_{\text{src}})$ to a receiver at $(\text{lat}_{\text{rcvr}}, \text{lon}_{\text{rcvr}}, z_{\text{grnd}})$.
`-wnl_wvfrm`	Compute the weakly non-linear waveform evolution along a single ray path.

- The `profile.met` file is expected to contain columns describing the atmosphere in the format:

$$z\,[\text{km}] \;\vdots\; T(z)\,[\text{K}] \;\vdots\; u(z)\left[\frac{\text{m}}{\text{s}}\right] \;\vdots\; v(z)\left[\frac{\text{m}}{\text{s}}\right] \;\vdots\; \rho_0(z)\left[\frac{\text{g}}{\text{cm}^3}\right] \;\vdots\; p_0(z)\,[\text{mbar}]$$

- Parameters are set using the format `parameter_name=value`, for example: `incl_step=1.0`. Possible parameters for each option are included below.

Parameters for propagation option		
Parameter	Description	Default Value
`incl_min`	Minimum inclination angle (0^o = horizontal)	0.5^o
`incl_max`	Maximum inclination angle (0^o = horizontal)	45.0^o
`incl_step`	Inclination step size	0.5^o
`inclination`	Set a *single* azimuth (`incl_max` = `incl_min`)	–
`az_min`	Minimum azimuth angle (North = 0^o, increases clockwise)	-90.0^o
`az_max`	Maximum azimuth angle (North = 0^o, increases clockwise)	-90.0^o
`az_step`	Azimuth step size	1.0^o
`azimuth`	Set a *single* azimuth (`az_max` = `az_min`)	–
`bounces`	Maximum # of bounces to compute (integer ≥ 0)	2
`src_lat`	Latitude of the source	30.0^o
`src_lon`	Longitude of the source	0.0^o
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`write_atmo`	Boolean to write the atmosphere specifications	false
`write_rays`	Boolean to write the ray paths	true

Parameters for eigenray search option		
Parameter	Description	Default Value
`incl_min`	Minimum inclination angle (0^o = horizontal)	0.5^o
`incl_max`	Maximum inclination angle (0^o = horizontal)	45.0^o
`bnc_min`	Minimum # of bounces to consider (integer)	0
`bnc_max`	Maximum # of bounces to consider (integer)	0
`bounces`	Exact # of bounces (`bnc_min` = `bnc_max`)	0
`src_lat`	Latitude of the source	30.0^o
`src_lon`	Longitude of the source	0.0^o
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_lat`	Latitude of the receiver	30.0^o
`rcvr_lon`	Longitude of the receiver	-2.5^o
`verbose`	Boolean to output details of the search	false
`iterations`	Maximum number of iterations in eigenray search	25
`damping`	Damping factor in Levenberg-Marquardt solver	1.0e-3
`tolerance`	Accepted distance from receiver location	0.1 km
`incl_step_min`	Smallest step size in search algorithm	1.0e-3^o
`incl_step_max`	Largest step size in search algorithm	0.1^o

Parameters for eigenray direct option		
Parameter	Description	Default Value
`incl_est`	Estimated inclination angle (0^o = horizontal)	15.0^o
`az_est`	Estimated azimuth angle (North = 0^o, increases clockwise)	atan2 (`y_rcvr`, `x_rcvr`)
`bounces`	Exact # of bounces to consider (integer ≥ 0)	0
`src_lat`	Latitude of the source	30.0^o
`src_lon`	Longitude of the source	0.0^o
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_lat`	Latitude of the receiver	30.0^o
`rcvr_lon`	Longitude of the receiver	-2.5^o
`verbose`	Boolean to output details of the search	false
`iterations`	Maximum number of iterations in eigenray search	25
`damping`	Damping factor in Levenberg-Marquardt solver	1.0e-3
`tolerance`	Accepted distance from receiver location	0.1 km

Parameters for weakly non-linear waveform option		
Parameter	Description	Default Value
`inclination`	Initial inclination angle of the ray	10^o
`azimuth`	Initial azimuth angle of the ray	90^o
`bounces`	# of bounces to compute (integer ≥ 0)	0
`src_x`	East/West offset of the source	0.0 km
`src_y`	North/South offset of the source	0.0 km
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
...	See Section 4.8	...

Output

`atmo.dat`
Contains the interpolate atmospheric profile. Columns are:

$$z\,[\mathrm{km}] \vdots c(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots u(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots v(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots \rho_0(z)\left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \vdots c_{\mathrm{eff}}\left(z,\varphi_0\right)\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$$

`file_name.raypaths.dat`
`file_name.eigenray-#.dat`
Contains the ray path information. Columns are:

$$z[\mathrm{km}] \vdots \mathrm{Lat}\ [\mathrm{deg}] \vdots \mathrm{Lon}\ [\mathrm{deg}] \vdots A_{\mathrm{geo.}}[\mathrm{dB}] \vdots A_{\mathrm{atmo.}}[\mathrm{dB}] \vdots t[\mathrm{sec}]$$

`file_name.arrivals.dat`
Contains the arrival information of each ground intercept for the `-prop` option:

$$\vartheta[\mathrm{deg}] \vdots \varphi[\mathrm{deg}] \vdots n_{\mathrm{bnc}} \vdots \mathrm{Lat}_0[\mathrm{km}] \vdots \mathrm{Lon}_0[\mathrm{km}] \vdots t[\mathrm{sec}] \vdots \nu\left[\frac{\mathrm{km}}{\mathrm{s}}\right] \vdots z_{\max}[\mathrm{km}] \vdots \vartheta_{\mathrm{inc.}}[\mathrm{deg}] \vdots \varphi_{\mathrm{back}}[\mathrm{deg}] \vdots A_{\mathrm{geo.}}[\mathrm{dB}] \vdots A_{\mathrm{atmo.}}[\mathrm{dB}]$$

Contains the run summary and arrival information for each eigenray for the `-eig_search` option (identical to 3D Stratified case)

2.4 Range Dependent Propagation

Description

The methods in `infraga-3d-rngdep` and `infraga-sph-rngdep` are identical to those in `infraga-3d` and `infraga-sph`, respectively, but allow for the more general case in which the atmosphere is not stratified. An example set of profiles and `.loc` files are included in the `examples` directory for reference and will be discussed in Section 3.2.

Usage

```
infraga-3d-rngdep  [option]  profile_id  nodes-x.loc  nodes-y.loc  [parameters]
```

```
infraga-sph-rngdep  [option]  profile_id  nodes-lat.loc  nodes-lon.loc  [parameters]
```

- Options, parameters, and output are identical to those for `infraga-3d`. for `-prop` and `-interactive`; however, the default values of the source and receiver positions are computed to be contained within the grid.

Parameters for range dependent Cartesian proapgation		
Parameter	Description	Default Value
`src_x`	East/West offset of the source	Midpoint of `nodes-x.loc` file
`src_y`	North/South offset of the source	Midpoint of `nodes-y.loc` file
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_x`	East/West offset of receiver	Midpoint of `nodes-x.loc` file – 250.0 km
`rcvr_y`	North/South offset of receiver	Midpoint of `nodes-y.loc` file

Parameters for range dependent Cartesian proapgation		
Parameter	Description	Default Value
`src_lat`	Latitude of the source	Midpoint of `nodes-lat.loc` file
`src_lon`	Longitude of the source	Midpoint of `nodes-lon.loc` file
`src_alt`	Altitude of the source (rel. sea level)	0.0 km
`rcvr_lat`	Latitude of the receiver	Midpoint of `nodes-lat.loc` file
`rcvr_lon`	Longitude of the receiver	Midpoint of `nodes-lon.loc` file + 5^o

3 Clarifications

3.1 Amplitude Format

The amplitude coefficient computed in the infraGA methods is defined in dB relative to spherical spreading at the source. That is, the amplitude approaches $10\log_{10}\left(\frac{1}{s}\right)$ for $s \downarrow 0$ with s measured in kilometers. This results in a 30 dB amplitude at 1 m and, assuming there is negligible refraction within the first kilometer, 0 dB amplitude at 1 km, which functions as an approximate reference distance for amplitudes in the far field.

3.2 Range Dependent Profile Interpolation

The range dependent interpolation scheme uses profiles specified by the definition

$$f_n(z) = f(x_i, y_j, z), \quad \text{with} \;\; n = i \times J + j.$$

For the global case, replace x with latitude and y with longitude. The multivariate interpolation scheme pre-computes vertical splines at each horizontal grid node and stores them in memory. The 3D interpolation is then evaluated using a bicubic interpolation in the horizontal which has been modified to guarantee continuous second order derivatives required for the continuity of the differential equations governing propagation. This interpolation scheme runs at approximately $\frac{1}{10}$ the speed of the stratified counterpart due to the added complexity of the range dependent propagation equations and the interpolation scheme. An example case is summarized below.

- For a 5×5 grid centered at $x, y = 0.0, 0.0$ with size $1,000 \times 1,000$ kilometers, you will need to define 25 meteorological files with names formatted as `profile0.met`, `profile1.met`, ..., `profile24.met`
- Each file designates the atmosphere at a particular x, y location defined by values in the `x_vals.loc` and `y_vals.loc` files. These files should have values listed in ascending order with mapping of the form:

`x_vals.loc`	`y_vals.loc`	Profile Mapping		
-500.0	-500.0	$x = -500.0$, $y = -500.0$	$\rightarrow$	`profile0.met`
-250.0	-250.0	$x = -500.0$, $y = -250.0$	$\rightarrow$	`profile1.met`
0.0	0.0	$x = -500.0$, $y = 0.0$	$\rightarrow$	`profile2.met`
250.0	250.0	$x = -500.0$, $y = 250.0$	$\rightarrow$	`profile3.met`
500.0	500.0	$x = -500.0$, $y = 500.0$	$\rightarrow$	`profile4.met`
		$x = -250.0$, $y = -500.0$	$\rightarrow$	`profile5.met`
		$x = -250.0$, $y = -250.0$	$\rightarrow$	`profile6.met`
		$\vdots$		$\vdots$
		$x = 500.0$, $y = 500.0$	$\rightarrow$	`profile24.met`

- The run summary will print the first few grid locations and which profiles are assigned to them, so use that output to confirm your grid matches that being constructed.

3.3 Running the Accelerated Methods

If you have OpenMPI, you can run the accelerated version of InfraGA. Usage of these methods is similar to the standard versions, but calling the method requires use of `mpirun` and a number of threads to be specified. The syntax to run the accelerated `infraga-3d` methods using 6 threads is:

```
mpirun -np 6 infraga-accel-3d [option] profile.met [parameters]
```

Only the initial search for eigenrays in `-eig_search` has been accelerated so that the `-eig_direct` option is not available. It should be noted that it is assumed that the accelerated methods are primarily used when only arrival information is required, so that the methods default to `write_rays=false`.

The eigenray search function in the accelerated methods can ingest lists of sources and/or receivers to distribute calculation of eigenrays for a network or from multiple sources. The formats of the source and receiver text files are expected to be columns of x, y, z and x, y, respectively, for Cartesian cases and latitude, longitude, altitude, and latitude, longitude for spherical coordinate runs. Note that the receiver is assumed to be on the ground surface so that only the source altitude can be defined. This method only works for cases in which the number of available CPU's is greater than the number of receiving arrays. If a very large number of receivers are needed, it's best to break them into separate lists such that the number of CPU's is at least 3 - 4 times the number of receivers in each search.

An additional option, `seq_srcs`, is used to further accelerate eigenray analysis when sources in the provided list are spatially similar and the solution for the previous eigenrays can be used as an initial guess for the current source. A reference distance, `seq_srcs_dr` is defined that determines when a new full eigenray search is performed. This method is still in development and may lead to missed eigenrays in some scenarios when `seq_srcs_dr` is set too large. The value defaults to 5 kms, which preliminary analyses indicate leads to a decent acceleration and few missed eigenrays.

4 Additional Parameters

In addition to those listed in the previous section, a number of global parameters can be used to modify the propagation scheme. These include `freq`, `z_grnd`, and `profile_format`. Additionally, while a Python script is included to convert G2S .bin files into range dependent files, a summary of the details of the expected inputs is presented here. Lastly, boolean values control the output of caustic points in the `-prop` option and the computation of amplitudes in the `-prop` and `-interactive` options.

4.1 Propagating Specific Frequencies and Modifying Absorption

Although weakly non-linear waveform evolution can be computed using the `-wnl_wvfrm` option, all methods can compute the Sutherland & Bass predicted absorption losses at a given frequency. The default frequency used to calculate attenuation is 0.1 Hz. By entering `freq=VALUE` other frequencies can be used. The absorption coefficient defaults to 0.3 (a scaling found to predict reasonable losses for thermospheric paths without including non-linear effects) but can be modified by `abs_coeff=VALUE`. This coefficient can be modified using the weakly non-linear solution in order to predict the waveform evolution without absorption losses.

4.2 Modifying the Ground Elevation

G2S profiles are defined with altitude relative to sea level. This often produces temperature and wind values extrapolated below ground level. For propagation at a known (average) elevation, one can modify the ground level with `z_grnd=VALUE`. Given such an option, the numerics will automatically set the source elevation to the specified ground level unless some higher elevation is entered.

It should be noted that this parameter is distinct from the `src_alt` value in that the ground surface elevation is used to determine the reflection point of ray paths, while the source can be at or above this elevation (in the case that the source is elevated above the ground surface, the inclination angle range should be modified to account for energy propagating downward, that is, `incl_min=-45.0` or similar). The methods automatically adjust the source height to ground level to satisfy `src_alt` $\geq$ `z_grnd` so that for a source at the ground surface, the user need only specify `z_grnd`.

4.3 Modifying the Profile Format

The expected format can be modified with the parameter statement `prof_format=TYPE`. Current options include the default, `zTuvdp`, and the NCPA format, `zuvwTdp`. As one might expect, this alternate profile format expects files with columns containing,

$$z\,[\mathrm{km}] \vdots u(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots v(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots w(z)\left[\frac{\mathrm{m}}{\mathrm{s}}\right] \vdots T(z)\,[\mathrm{K}] \vdots \rho_0(z)\left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \vdots p_0(z)\,[\mathrm{mbar}]\,.$$

Similarly, if the sound speed is known directly, one can specify `prof_format=zcuvd` where the sound speed and winds are all defined in meters-per-second and density is defined in g/cm^3 as in other formats. A script is included in the `scripts` directory to extract usable specifications or grids of specification from netCDF format ECMWF files. Usage of that script can be viewed by running it without arguments.

4.4 Writing the Caustic Points to File

When using the `-prop` option, an additional parameter is available: `write_caustics`. This is defaulted to `false` but when set to `true` will create a number of files entitled `file_name.caustics-#.dat`. These files contain the locations at which the Jacobian determinant changes sign which indicates passage through a caustic. The numerical value indicates the path between reflection `#` and `#+1`. Columns of the file contain:

$$x[\mathrm{km}] \vdots y[\mathrm{km}] \vdots z[\mathrm{km}] \vdots t[\mathrm{sec}]$$

Computation of these points helps to identify the caustic surface structure and validate that the geometric spreading factor is being accurately computed.

4.5 Disabling the Calculation of Amplitudes

Computation of the Jacobian terms requires solving a number of additional equations. When the amplitude is not needed, disabling this computation increases the ray path computation speed by a factor of at least 3 (since many second order derivative terms are no longer calculated as well). To disable the computation of the Jacobian coefficients, set `calc_amp=false` in the command line parameter list. This option is only available in the `-prop` runs since the eigenray methods require access to the auxiliary parameters used in computing amplitude to identify corrections to the launch angles for the eigenray search.

4.6 Modifying the Propagation Region

In some cases, it is useful to modify the extent of the propagation region. In the case of a stratified atmosphere, the default propagation region extends to the upper limit of the interpolated atmosphere file and to a range of 10,000 kilometers. In many cases, in addition to limiting the number of ground reflections, limiting the propagation range and altitude can be an efficient way to improve computation time. Listed in the table below are available options in each method to modify the propagation region. In the case of range dependent propagation, the default horizontal limits are set by the interpolated atmosphere grid.

Propagation Region Modifications					
	`2d`	`3d`	`3d-rngdep`	`sph`	`sph-rngdep`
`alt_max`	✓	✓	✓	✓	✓
`rng_max`	✓	✓	✓	✓	✓
`x_min`		✓	✓		
`x_max`		✓	✓		
`y_min`		✓	✓		
`y_max`		✓	✓		
`lat_min`				✓	✓
`lat_max`				✓	✓
`lon_min`				✓	✓
`lon_max`				✓	✓

The default altitude maximum is set by the provided atmosphere specification file and the default range maximum is 2,000 kilometers. In the range dependent methods, the default x, y, latitude, and longitude limits are set by the specified atmosphere region in the `.loc` files and in the general cases they default to $\pm 2,000$ km for the `3d` methods and all possible latitudes ($\pm 90^o$) and longitudes ($\pm 180^o$) for the `sph` methods.

4.7 Incorporating Ground Topography

The propagation of infrasound over non-flat ground can be modeled by specifying a topography file. The file format expected for each propagation scheme is summarized in the table below. It should be noted that the inclusion of topography in geometric acoustics will not model the effects of diffraction of sound around features, but will account for the variable ground elevation and some of the scattering of energy due to the the ground slope. It should be noted that the auxiliary parameters vary with the 2nd order derivative of the ground surface, so that focusing and spreading of rays due to the ground surface curvature is included in the computation of the geometric attenuation factor.

Topography File Format	
`2d`	$r \vdots z_g(r)$
`3d`	$x \vdots y \vdots z_g(x, y)$
`sph`	$\theta \vdots \phi \vdots z_g(\theta, \phi)$

The specified topography file is expected to be an ascii specification of the ground elevation (relative to sea level). A Python script is included to create such files on a line or grid, but require the user download the necessary ETOPO1 file (available at `https://www.ngdc.noaa.gov/mgg/global/`). A script is included in the `scripts` directory that will extract lines for `2d` propagation or grids for `3d` and `sph` simulations. Usage of the script can be viewed by running it without arguments.

Note that inclusion of topography can significantly increase computation time due to the additional evaluations of the ground interpolation for ground intercept checks and boundary layer no-slip condition. Additionally, the eigenray methods are very slow to converge (if they converge at all) and require a significantly smaller `incl_step_max` when including topography in order to identify an approximate eigenray.

4.8 Computing waveforms along ray paths

For all non-accel methods, a method, `-wnl_wvfrm`, is available to compute waveform evolution along individual ray paths using the method developed by Lonzaga *et al.* (2014). Some of the parameters for running this method were included in the previous discussion, others related to the waveform definitions and solver are below. When using a waveform from a specified file it is important to ensure the ingested waveform is pre-processed to tapered for the FFT and zero padded to allow for any waveform stretching.

Parameters for weakly non-line waveform calculations		
Parameter	Description	Default Value
`wvfrm_file`	File containing t, $p(t)$ at the reference location	-
`wvfrm_opt`	Option for the built-in waveforms: `impulse`, `Uwave`, `Nwave`	`impulse`
`wvfrm_p0`	Peak amplitude (for built-in waveform)	10.0 Pa
`wvfrm_t0`	Time scale (for built-in waveforms)	1.0 s
`wvfrm_alpha`	Shaping parameter (for built-in waveforms)	2.0
`wvfrm_ref`	Reference distance for initial waveform	1.0 km
`wvfrm_out_step`	Frequency at which to output scaled waveform along the path	none
`wvfrm_ds`	Reference step size along the ray for waveform calculation	0.5 km
`wvfrm_N_rise`	Points sampled over the rise time when using built-in waveform	10
`wvfrm_N_period`	Points-per-period when using built-in waveform	40
`wvfrm_len`	Length of the waveform when using a built-in waveform	2^{13}
`wvfrm_filt1_g`	Filter 1 cutoff frequency (rel. to Nyquist)	0.5
`wvfrm_filt1_n1`	Filter 1 exponential coefficient, n_1	4.0
`wvfrm_filt1_n2`	Filter 1 exponential coefficient, n_2	0.5
`wvfrm_filt2_g`	Filter 2 cutoff frequency (rel. to Nyquist)	0.75
`wvfrm_filt2_n1`	Filter 2 exponential coefficient, n_1	2.0
`wvfrm_filt2_n21`	Filter 2 exponential coefficient, n_2	1.0

The (non-normalized) waveforms are defined as:

$$p_{\text{impulse}}(t; t_0, \alpha) = \begin{cases} x^{\alpha}\left(1 - \frac{x}{1+\alpha}\right)e^{-x} & x > 0 \\ 0 & \text{else} \end{cases}, \qquad x = \frac{t}{t_0}$$

$$p_{\text{Uwave}}(t; t_0, \alpha) = \frac{\partial}{\partial t} p_{\text{impulse}}(t; t_0, \alpha), \qquad p_{\text{Nwave}}(t; t_0) = \begin{cases} \frac{t}{t_0} & |t| \leq t_0 \\ 0 & \text{else} \end{cases}$$

Generation of high frequency energy near the Nyquist causes instabilities in the solution that are often not entirely damped out by inclusion of Sutherland & Bass absorption losses. A pair of filters are utilized in suppressing Gibbs phenomenon in the non-linear waveform analysis to prevent excess energy from building up at the Nyquist frequency and causing errors. One filter is applied to the non-linear correction added to the FFT of the signal, and a second is applied to the waveform itself at each analysis step. Varying these filter parameters as well as the step size in `wvfrm_ds` for the solver are often needed to obtain usable waveform predictions. The form of the filters are below,

$$\mathcal{F}(f; \gamma, n_1, n_2) = \frac{1}{\left(1 + \left(\frac{f}{\gamma f_{\text{nyq}}}\right)^{n_1}\right)^{n_2}}$$

5 Plotting the Results

Included here are several short gnuplot scripts which are useful for displaying results. In all cases, you'll need to attached the appropriate prefix to the raypath.dat and results.dat files to be plotted.

2D Raypaths

```
    set pm3d map
    set palette defined (0 "white", 1 "blue", 2 "green", 3 "yellow", 4 "orange", 5 "red")
    set cbrange[-150:-50]
    splot '{...}.raypaths.dat' u 1:2:($3 + $4) w lines lw 1 palette
```

2D Results

```
    set pm3d map
    set palette defined (0 "white", 1 "blue", 2 "green", 3 "yellow", 4 "orange", 5 "red")
    set cbrange[-100:-50]
    splot '{...}.arrivals.dat' u ($5 - $4 / 0.34):4:($9 + $10) with points pt 7 ps 0.75 palette
```

3D and Spherical Raypaths

```
    rng(x, y) = sqrt(x**2 + y**2)
    set pm3d map
    set palette defined (0 "white", 1 "blue", 2 "green", 3 "yellow", 4 "orange", 5 "red")
    set cbrange[-150:-50]
    splot '{...}.raypaths.dat' u 1:3:($4 +$5) w lines lw 1 palette
splot '{...}.raypaths.dat' u 2:3:($4 +$5) w lines lw 1 palette
```

3D and Spherical Results

```
    set pm3d map
    set palette defined (0 "white", 1 "blue", 2 "green", 3 "yellow", 4 "orange", 5 "red")
    set cbrange[-100:-50]
    splot '{...}.arrivals.dat' u 4:5:($11 + $12) with points palette
    set palette defined (0 "red", 12 "red", 15 "green", 70 "green", 80 "blue", 140 "blue")
    set cbrange[0:140]
    splot '{...}.arrivals.dat' u 4:5:8 with points palette
```

Note: swap `4:5` to `5:4` for spherical coordinate cases to put longitude on the horizontal axis.

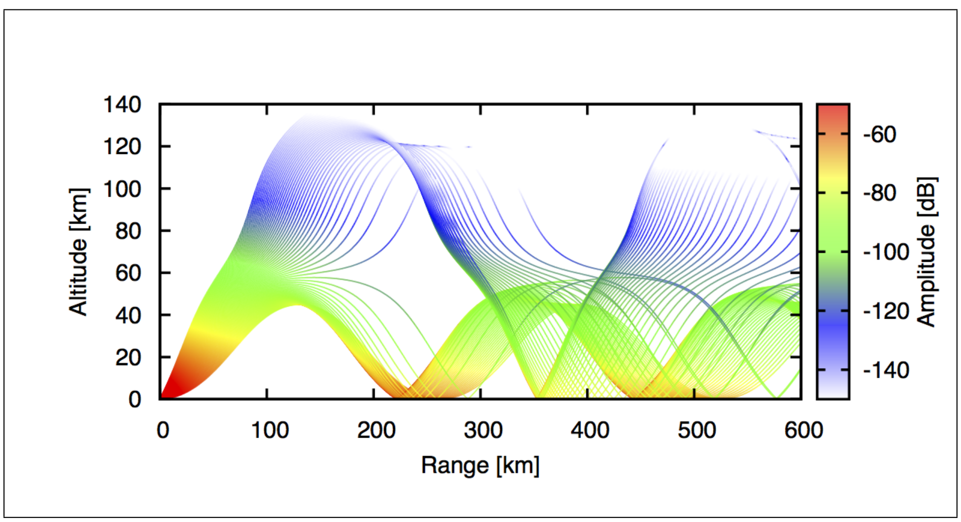

Figure 1: Example ray path figure generated with GNUPlot commands.

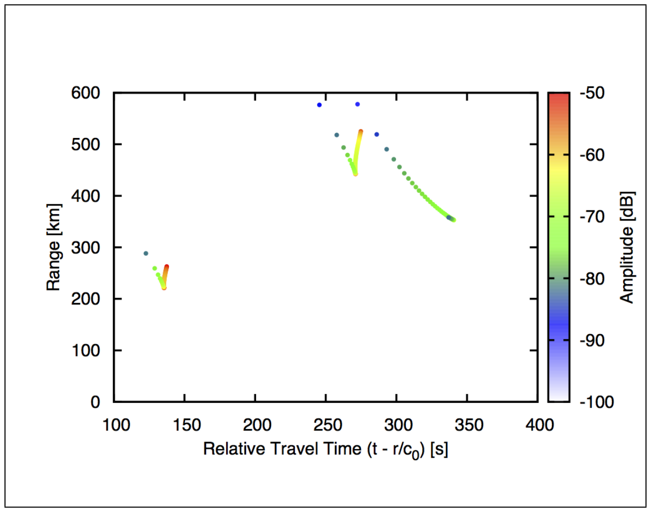

Figure 2: Relative travel time figure generated with GNUPlot commands.

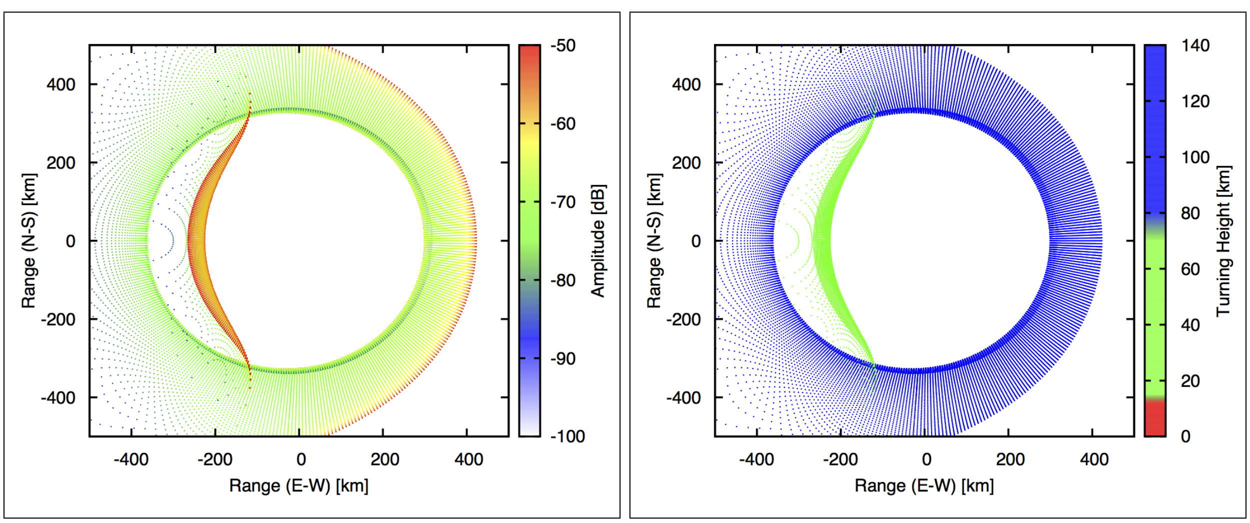

Figure 3: Arrival amplitudes and turning heights for 3D propagation. Similar results for global case.

6 Mathematical Background - What Does infraGA Do?

The propagation of acoustic energy can be described by a linear perturbation of the fluid mechanics equations. The linear order continuity, Euler, and state equations for an inhomogeneous, moving medium have the forms,

$$\frac{D\rho}{Dt} + \rho \vec{\nabla} \cdot \vec{v}_0 + \vec{\nabla} \cdot (\rho_0 \vec{v}) = 0, \tag{1a}$$

$$\frac{D\vec{v}}{Dt} + \left(\vec{v} \cdot \vec{\nabla}\right) \vec{v}_0 = -\frac{1}{\rho_0} \vec{\nabla} p + \rho \vec{\nabla} \frac{p_0}{\rho_0^2}, \tag{1b}$$

$$\vec{v} \cdot \vec{\nabla} p_0 + \frac{Dp}{Dt} = c^2 \left[\vec{v} \cdot \vec{\nabla} \rho_0 + \frac{D\rho}{Dt} \right] + \left(c^2\right)' \vec{v}_0 \cdot \vec{\nabla} \rho_0, \tag{1c}$$

where subscript 0's denote ambient quantities that vary in space and those without subscripts denote acoustic perturbations that vary in space and time. The approximation of geometric acoustics is constructed by expanding each variable with a spatially varying phase, $e^{ik_0 \lambda(\vec{x})}$, and Debye series, $\sum \frac{\mathcal{P}_j(\vec{x})}{(ik_0)^j}$. The phase function, $\psi\left(\vec{x}\right)$, is termed the Eikonal and its solution provides information about the deformation of surfaces of constant phase. Expanding each linear variable,

$$\begin{pmatrix} p \\ \vec{v} \\ \rho \\ \left(c^2\right)' \end{pmatrix} = e^{ik_0 \lambda(\vec{x})} \sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \begin{pmatrix} \mathcal{P}_j\left(\vec{x}\right) \\ \vec{\mathcal{V}}_j\left(\vec{x}\right) \\ \mathcal{D}_j\left(\vec{x}\right) \\ \mathcal{C}_j\left(\vec{x}\right) \end{pmatrix}. \tag{2}$$

The linearized fluid mechanics equations can then be written as,

$$\sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \Big\{ - ik_0 \mathcal{D}_j \left(c_0 - \vec{v}_0 \cdot \vec{\psi}\right) + \vec{v}_0 \cdot \vec{\nabla} \mathcal{D}_j + \mathcal{D}_j \vec{\nabla} \cdot \vec{v}_0 + \rho_0 \vec{\nabla} \cdot \vec{\mathcal{V}}_j + \rho_0 ik_0 \vec{\mathcal{V}}_j \cdot \vec{\psi} + \vec{\mathcal{V}}_j \cdot \vec{\nabla} \rho_0 \Big\} = 0, \tag{3a}$$

$$\sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \left\{ -ik_0 \vec{\mathcal{V}}_j \left(c_0 - \vec{v}_0 \cdot \vec{\psi}\right) + \vec{v}_0 \cdot \vec{\nabla} \vec{\mathcal{V}}_j + \vec{\mathcal{V}}_j \cdot \vec{\nabla} \vec{v}_0 \right\} = \sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \left\{ -\frac{ik_0}{\rho_0} \mathcal{P}_j \vec{\psi} - \frac{1}{\rho_0} \vec{\nabla} \mathcal{P}_j + \mathcal{D}_j \vec{\nabla} \frac{p_0}{{\rho_0}^2} \right\}, \tag{3b}$$

$$\begin{aligned} &\sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \left\{ \vec{\mathcal{V}}_j \cdot \vec{\nabla} p_0 - ik_0 \mathcal{P}_j \left(c_0 - \vec{v}_0 \cdot \vec{\psi}\right) + \vec{v}_0 \cdot \vec{\nabla} \mathcal{P}_j \right\} \\ &\qquad = \sum_{j=0}^{\infty} \frac{1}{\left(ik_0\right)^j} \Big\{ c^2 \left[\vec{\mathcal{V}}_j \cdot \vec{\nabla} \rho_0 - ik_0 \mathcal{D}_j \left(c_0 - \vec{v}_0 \cdot \vec{\psi}\right) + \vec{v}_0 \cdot \vec{\nabla} \mathcal{D}_j \right] + \mathcal{C}_j \vec{v}_0 \cdot \vec{\nabla} \rho_0 \Big\}, \end{aligned} \tag{3c}$$

where we've defined $\vec{\psi} = \vec{\nabla} \lambda$. Collecting terms in powers of k_0, the leading order contributions require

$$\left(1 - \frac{\vec{v}_0 \cdot \vec{\psi}}{c_0}\right) \mathcal{D}_0 = \frac{\rho_0}{c_0} \vec{\mathcal{V}}_0 \cdot \vec{\psi}, \tag{4a}$$

$$\left(1 - \frac{\vec{v}_0 \cdot \vec{\psi}}{c_0}\right) \vec{\mathcal{V}}_0 = \frac{1}{\rho_0 c_0} \mathcal{P}_0 \vec{\psi}, \tag{4b}$$

$$\mathcal{P}_0 = c^2 \mathcal{D}_0, \tag{4c}$$

which can be combined to obtain the Eikonal Equation,

$$\psi^2 = \frac{c_0^2}{c^2\left(\vec{x}\right)} \left[1 - \frac{\vec{v}_0\left(\vec{x}\right) \cdot \vec{\psi}}{c_0} \right]^2. \tag{5}$$

In addition to the ray path geometry defined by the Eikonal Equation, higher order terms in the expansion provide a means to estimate ray spreading and geometric attenuation. Taking the terms in the expansion proportional to k_0,

$$\left(1 - \frac{\vec{v}_0 \cdot \vec{\psi}}{c_0}\right) \vec{\mathcal{V}}_1 - \frac{1}{\rho_0 c_0} \mathcal{P}_1 \vec{\psi} = \frac{1}{c_0} \left[\vec{v}_0 \cdot \vec{\nabla} \vec{\mathcal{V}}_0 + \vec{\mathcal{V}}_0 \cdot \vec{\nabla} \vec{v}_0 + \frac{1}{\rho_0} \vec{\nabla} \mathcal{P}_0 - \frac{\mathcal{D}_0}{\rho_0{}^2} \vec{\nabla} p_0 \right] = \vec{b}, \tag{6a}$$

$$\left(1 - \frac{\vec{v}_0 \cdot \vec{\psi}}{c_0}\right) \mathcal{D}_1 - \frac{\rho_0}{c_0} \vec{\psi} \cdot \vec{\mathcal{V}}_1 = \frac{1}{c_0} \vec{\nabla} \cdot \left(\mathcal{D}_0 \vec{v}_0 + \rho_0 \vec{\mathcal{V}}_0 \right) = b_1, \tag{6b}$$

$$\mathcal{P}_1 - c^2 \mathcal{D}_1 = \frac{1}{c\psi} \left[\vec{\mathcal{V}}_0 \cdot \vec{\nabla} p_0 + \vec{v}_0 \cdot \vec{\nabla} \mathcal{P}_0 - c^2 \vec{v}_0 \cdot \vec{\nabla} \mathcal{D}_0 - \frac{\mathcal{P}_0}{c^2} \vec{v}_0 \cdot \vec{\nabla} c^2 - c^2 \vec{\mathcal{V}}_0 \cdot \vec{\nabla} \rho_0 \right] = b_2, \tag{6c}$$

Using the Eikonal Equation condition, these equations can be combined in a manner which goes to zero,

$$\begin{cases} \psi \vec{\mathcal{V}}_1 - \frac{1}{\rho_0 c_0} \mathcal{P}_1 \vec{\psi} = \vec{b} \\ \psi \mathcal{D}_1 - \frac{\rho_0}{c_0} \vec{\psi} \cdot \vec{\mathcal{V}}_1 = b_1 \\ \quad \mathcal{P}_1 - c^2 \mathcal{D}_1 = b_2 \end{cases} \rightarrow \quad \frac{c_0 \rho_0}{\psi} \vec{\psi} \cdot \vec{b} + c_0 c b_1 + \psi b_2 = 0. \tag{7}$$

Solving this condition leads to the transport equation,

$$\vec{\nabla} \cdot \left(\mathcal{P}_0^2 \vec{c}_g \right) = \mathcal{P}_0^2 \vec{c}_g \cdot \vec{\nabla} \ln \left(\rho_0 c^3 \psi \right), \tag{8}$$

and the resulting amplitude term is defined in terms of the Jacobian, $D(s, \vartheta, \varphi)$ where s, ϑ, and φ are the ray length, initial inclination angle, and initial azimuthal angle of the ray path respectively, that describes the coordinate transformation between Cartesian and ray coordinates,

$$\mathcal{P}_0(s, \vartheta, \varphi) = \frac{1}{4\pi} \left| \frac{\rho_0(s)\, \psi(s)\, c^3(s)}{\rho_0(0)\, \psi(0)\, c^3(0)} \frac{c_g(0) \cos\vartheta}{c_g(s)\, D(s, \vartheta, \varphi)} \right|^{\frac{1}{2}}. \tag{9}$$

Spherical spreading at the source has been assumed so that $\mathcal{P}_0(s, \vartheta, \varphi)|_{s \downarrow 0} = \frac{1}{4\pi s^2}$ and $D(s, \vartheta, \varphi)|_{s \downarrow 0} = s^2 \cos\vartheta$,

The Eikonal Equation can be used to define a Hamiltonian, $H\left(\vec{x}, \vec{\psi}\right) = 0$ and the Hamilton-Jacobi relations used to define equations governing ray paths,

$$\frac{\partial \vec{x}}{\partial \tau} = \frac{\partial H}{\partial \vec{\psi}}, \qquad \frac{\partial \vec{\psi}}{\partial \tau} = -\frac{\partial H}{\partial \vec{x}}. \tag{10}$$

For the methods in infraGA, the parameter τ is replaced by ray length, s, such that $|d\vec{x}| = ds$. The transport coefficient depends on the Jacobian determinant which is defined by the variation between coordinate systems. Denoting the initial launch inclination and azimuth as ϑ and φ, respectively, one has,

$$D(x, y, z; s, \vartheta, \varphi) = \left| \frac{\partial(x, y, z)}{\partial(s, \vartheta, \varphi)} \right| = \mathbf{det} \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial \vartheta} & \frac{\partial x}{\partial \varphi} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial \vartheta} & \frac{\partial y}{\partial \varphi} \\ \frac{\partial z}{\partial s} & \frac{\partial z}{\partial \vartheta} & \frac{\partial z}{\partial \varphi} \end{pmatrix} \tag{11}$$

As detailed in Blom & Waxler (2012), the s derivatives can be defined directly from the Eikonal Equation condition, but the ϑ and φ derivatives require the introduction of auxiliary parameters, $\mathcal{X}^{(\vartheta)} = \frac{\partial x}{\partial \vartheta}$, $\mathcal{X}^{(\varphi)} = \frac{\partial x}{\partial \varphi}$ with similar parameters defined for $\mathcal{Y}$ and $\mathcal{Z}$.

6.1 Propagation in two-dimensions using the effective sound speed

In the case of the effective sound speed approximation, one re-defines $c \rightarrow c + \vec{v}_0 \cdot \hat{\psi}_\perp$ (adding the wind in the direction of propagation to the adiabatic sound speed) and $\vec{v} = 0$ in the relations. This reduces the Eikonal to, $\psi^2 = \frac{c_0^2}{c^2}$, and the propagation relations become simply,

$$\frac{\partial \vec{x}}{\partial s} = \frac{c_0}{c}\vec{\psi}, \qquad \frac{\partial \psi_j}{\partial s} = -\frac{c_0}{c^2}\frac{\partial c}{\partial x_j}, \tag{12a}$$

6.2 Propagation in three-dimensions using an inhomogeneous moving medium

The differential equations describing the geometric ray paths in an arbitrary moving medium can be found from the Eikonal derived above,

$$\frac{\partial \vec{x}}{\partial s} = \frac{\vec{c}_g}{c_g}, \quad \vec{c}_g = c\frac{\vec{\psi}}{\psi} + \vec{v}_0 \tag{13a}$$

$$\frac{\partial \psi_j}{\partial s} = -\frac{1}{c_g}\left[\psi \frac{\partial c}{\partial x_j} + \vec{\psi} \cdot \frac{\partial \vec{v}_0}{\partial x_j}\right]. \tag{13b}$$

This is the equation set used in the `infraga-3d` methods.

6.3 Propagation in spherical coordinates

The eikonal solution in spherical coordinates has been derived and can be summarized as,

$$\frac{\partial u_j}{\partial s} = \mathcal{G}_j \frac{c_{g,j}}{c_g}, \qquad c_{g,j} = c\frac{\psi_j}{\psi} + v_j, \tag{14a}$$

$$\frac{\partial \psi_j}{\partial s} = -\frac{\mathcal{G}_j}{c_g}\left(\psi \frac{\partial c}{\partial u_j} + \sum_k \psi_k \frac{\partial v_k}{\partial u_j} + \mathcal{T}_j\right), \tag{14b}$$

where geometric scaling coefficients and corrective terms for the spatial variability of the spherical coordinate unit vectors produce,

$$\mathcal{G}_r = 1, \quad \mathcal{G}_\theta = \frac{1}{r}, \quad \mathcal{G}_\phi = \frac{1}{r\cos\theta}, \tag{14c}$$

$$\mathcal{T}_r = \frac{1}{r}\left(\psi_\theta c_{g,\theta} + \psi_\phi c_{g,\phi}\right), \tag{14d}$$

$$\mathcal{T}_\theta = \left(\psi_r v_\theta - \psi_\theta v_r\right) - \left(\psi_r c_{g,\theta} - \psi_\phi c_{g,\phi}\tan\theta\right), \tag{14e}$$

$$\begin{aligned}\mathcal{T}_\phi = {} & \left(\psi_r v_\phi - \psi_\phi v_r\right)\cos\theta \\ & + \left(\psi_\theta v_\phi - \psi_\phi v_\theta\right)\sin\theta \\ & - c_{g,\phi}\left(\psi_r \cos\theta + \psi_\theta \sin\theta\right).\end{aligned} \tag{14f}$$

This is the equation set used in the `infraga-sph` methods.

6.4 Eigenray methods

Eigenrays are identified using the auxiliary parameters defined in order to compute the Jacobian components needed to calculate geometric spreading. Considering the arrival location of a ray path in 3D,

$$x_0\left(\theta + \delta\theta, \phi + \delta\phi\right) = x_0\left(\theta, \phi\right) + \frac{\partial x_0}{\partial \theta}\delta\theta + \frac{\partial x_0}{\partial \phi}\delta\phi + O\left(\delta^2\right), \tag{15a}$$

$$y_0\left(\theta + \delta\theta, \phi + \delta\phi\right) = y_0\left(\theta, \phi\right) + \frac{\partial y_0}{\partial \theta}\delta\theta + \frac{\partial y_0}{\partial \phi}\delta\phi + O\left(\delta^2\right), \tag{15b}$$

This can be written more compactly as,

$$\begin{pmatrix} \delta x_0 \\ \delta y_0 \end{pmatrix} = \begin{pmatrix} \frac{\partial x_0}{\partial \theta} & \frac{\partial x_0}{\partial \phi} \\ \frac{\partial y_0}{\partial \theta} & \frac{\partial y_0}{\partial \phi} \end{pmatrix} \begin{pmatrix} \delta \theta \\ \delta \phi \end{pmatrix}, \tag{15c}$$

or simply,

$$\delta \vec{x}_0 = \boldsymbol{\mathcal{D}}_0 \, \delta \vec{\vartheta}. \tag{15d}$$

From this linear approximation, a Levenberg-Marquardt algorithm can be constructed,

$$\delta \vec{\vartheta} = \left(\boldsymbol{\mathcal{D}}_0 + \lambda \, \mathrm{diag} \left(\boldsymbol{\mathcal{D}}_0 \right) \right)^{-1} \delta \vec{x}_0, \tag{15e}$$

that will identify the changes in ray launch angles, $\delta \vec{\vartheta}$ needed to shift the arrival location by some distance, $\delta \vec{x}_0$. This algorithm is utilized as a stand alone method in `-eig_direct` and as the precision search step in `-eig_search` where a preliminary inclination/range search is used to identify initial solutions near eigenrays.

6.5 Ground reflections and topography

The reflection conditions for including topography are modified so that the eikonal vector components along the ground surface are conserved and the normal component to the ground changes sign,

$$\vec{\psi}_{\text{refl}} \cdot \hat{n}_{\text{grnd}} = -\vec{\psi} \cdot \hat{n}_{\text{grnd}}, \quad \vec{\psi}_{\text{refl}} \times \hat{n}_{\text{grnd}} = \vec{\psi} \times \hat{n}_{\text{grnd}}. \tag{16}$$

The resulting reflection conditions are then defined by relating the ground normal to the surface derivative,

$$\psi_x \left(s_0 + 0^+, \vartheta, \varphi \right) = \mathcal{C}_1^{(x)} \psi_{x,0} + \mathcal{C}_2^{(x)} \left(\psi_{z,0} - \psi_{y,0} \frac{\partial z_g}{\partial y} \right), \tag{17a}$$

$$\psi_y \left(s_0 + 0^+, \vartheta, \varphi \right) = \mathcal{C}_1^{(y)} \psi_{y,0} + \mathcal{C}_2^{(y)} \left(\psi_{z,0} - \psi_{x,0} \frac{\partial z_g}{\partial x} \right), \tag{17b}$$

$$\psi_z \left(s_0 + 0^+, \vartheta, \varphi \right) = -\mathcal{C}_1^{(z)} \psi_{z,0} + \mathcal{C}_2^{(x)} \psi_{x,0} + \mathcal{C}_2^{(y)} \psi_{y,0}. \tag{17c}$$

where,

$$\mathcal{C}_1^{(x)} = \frac{1 - \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2}{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2}, \qquad \mathcal{C}_2^{(x)} = \frac{2 \frac{\partial z_g}{\partial x}}{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2},$$

$$\mathcal{C}_1^{(y)} = \frac{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 - \left(\frac{\partial z_g}{\partial y} \right)^2}{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2}, \qquad \mathcal{C}_2^{(y)} = \frac{2 \frac{\partial z_g}{\partial y}}{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2},$$

$$\mathcal{C}_1^{(z)} = \frac{1 - \left(\frac{\partial z_g}{\partial x} \right)^2 - \left(\frac{\partial z_g}{\partial y} \right)^2}{1 + \left(\frac{\partial z_g}{\partial x} \right)^2 + \left(\frac{\partial z_g}{\partial y} \right)^2}.$$

6.6 Weakly non-linear waveform calculation

The waveform evolution is computed using the methods developed by Lonzaga *et al.* (2015) using a Heun's solver (RK2). The equations being solved are,

$$\frac{\partial u}{\partial s} = \tilde{\beta} u \frac{\partial u}{\partial \tau}, \quad \tilde{\beta} \left(s \right) = \beta \frac{p_0}{\rho_0 c_0^2} \frac{\psi_0 c_0}{c_{g,0} c_{\text{src}}} \sqrt{\frac{D_0 \rho_0 c_{g,0}^3}{D \rho c_g^3} \frac{c \psi^3}{c_0 \psi_0^3}}, \tag{18a}$$

$$u \left(s, \tau \right) = \frac{p \left(s, \tau \right)}{p_{\text{ref}}} \sqrt{\frac{\rho_0 c_0^3 \psi_0}{\rho c^3 \psi} \frac{c_g D}{c_{g,0} D_0}}, \tag{18b}$$

where subscript zeros denote evaluation at some reference point, $s = s_0$, along the ray path. The variable step size in the solver is defined as $ds = ds_0 / \left(\pi \tilde{\beta}(s) \max(\mathcal{U}(s,f))\right)$, where $\mathcal{U}(s,f)$ is the FFT of $u(s,t)$ along the ray path and ds_0 is defined in the code as `wvfrm_ds`.

In cases for which little energy is expected hear the Nyquist frequency, a value of $ds_0 \sim 1.0$ can be used; however, for source waveforms with high frequency content (e.g., a blast wave) or propagation paths extending into the upper atmosphere where rarefaction leads to strong relatively strong non-linear effects and generation of high frequency energy, a value of ~ 0.1 might be required. Best practice is to vary the value of `wvfrm_ds` to be sure your analysis has converged.