Package ‘AtmRay’

February 19, 2015

Type Package
Title Acoustic Traveltime Calculations for 1-D Atmospheric Models
Version 1.31
Date 2012-09-28
Author Jake Anderson
Maintainer Jake Anderson <ajakef@gmail.com>
Description Calculates acoustic traveltimes and ray paths in 1-D, linear atmospheres. Later versions will support arbitrary 1-D atmospheric models, such as radiosonde measurements and standard reference atmospheres.
Suggests RSEIS
License GPL
LazyLoad yes
Repository CRAN
Date/Publication 2013-03-01 22:39:55
NeedsCompilation yes

R topics documented:

A2P.lin ................................................................. 2
CheckAtm.lin ........................................................... 3
MakeArrivals.lin ...................................................... 4
MakeAtmList.lin ...................................................... 5
meshgrid .............................................................. 6
meshgridn ............................................................ 7
P4X.lin ............................................................... 7
P4X.lin_no_c ......................................................... 8
PlotAtm.lin .......................................................... 9
Prop.lin .............................................................. 10
Prop.lin_no_c ....................................................... 11
RESCALE .......................................................... 12
sky.colors ......................................................... 13
A2P.lin

Convert Between Incidence Angle and Ray Parameter

Description

Given an atmosphere and elevation, converts incidence angle to ray parameter, or vice-versa.

Usage

A2P.lin(angle, z, az, ATM)
P2A.lin(p, z, az, ATM)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>Ray parameter (s/m)</td>
</tr>
<tr>
<td>angle</td>
<td>Incidence angle (from vertical) in degrees</td>
</tr>
<tr>
<td>z</td>
<td>Elevation (m)</td>
</tr>
<tr>
<td>az</td>
<td>Compass azimuth (degrees)</td>
</tr>
<tr>
<td>ATM</td>
<td>Linear atmosphere</td>
</tr>
</tbody>
</table>

Value

Ray parameter (s/m) or incidence angle (degrees).

Author(s)

Jake Anderson

Examples

ATM = CheckAtm.lin()
A2P.lin(50, 50, 0, ATM)
P2A.lin(0.002321347, 50, 0, ATM)
Check Linear Atmosphere

Description

Verifies that an atmosphere contains all required elements and fills in missing elements with default values.

Usage

CheckAtm.lin(ATM = list())

Arguments

ATM List that may include characteristics of an atmosphere

Details

A linear atmosphere variable must contain the elements described in the "Value" section. Default values for these are as follows: z0: 0 c0: 330 wx0: 0 wy0: 0 rho0: 1.2929 * exp(-ATM$z0/6800) gc: -10^9 gwx: 0 gwy: 0 grho: -0.0001901058 * exp(-ATM$z0/6800)

Other functions may encounter problems when working with an effective sound speed gradient of zero.

Value

List including the following elements:

z0 Elevation of intercept layer (m)
c0 Intrinsic sound speed at intercept layer (m/s)
wx0 Zonal (east-west) wind at intercept layer (m/s)
wy0 Meridional (north-south) wind at intercept layer (m/s)
rho0 Density at intercept layer (kg/m^3)
gc Vertical intrinsic sound speed gradient (1/s)
gwx Vertical zonal wind gradient (1/s)
gwy Vertical meridional wind gradient (1/s)
grho Vertical density gradient (kg/m^4)

Author(s)

Jake Anderson
Examples

# quickly make a new default atmosphere
ATM = CheckAtm.lin()

# fill in missing values for an existing atmosphere
ATM = list(c0 = 343, gc = -0.006)
ATM = CheckAtm.lin(ATM)

Description

Calculates arrival time and arrival amplitude of many sources to produce an arrival function.

Usage

MakeArrivals.lin(xsL, ysL, zsL, xrL, yrL, zrL, dtL, ntL, timingL, ATM = CheckAtm.lin(list()))

Arguments

- xs: source location abscissae (m)
- ys: source location ordinates (m)
- zs: source location elevations (m)
- xr: receiver location abscissae (m)
- yr: receiver location ordinates (m)
- zr: receiver location elevations (m)
- dt: time interval (s)
- nt: number of time steps to evaluate
- timing: source times (s)
- ATM: linear atmosphere

Details

This is suitable for calculating arrival functions for spatially distributed acoustic sources. Each of sx, sy, and sz should be the same length, and each of rx, ry, and rz should be the same length.

Value

Returns a matrix of dimension nt x length(rx), with rows corresponding to time steps and columns to receivers.

Author(s)

Jake Anderson
Example

MakeArrivals.lin(xs = c(100, 150), ys = c(100, 150), zs = c(100, 150), xr = 0, yr = 0, zr = 0, dt = 0.01, nt = 100, 
MakeAtmList.lin

Make List of Linear Atmospheres

Description

Given a set of values that must be taken by each field in a linear atmosphere, returns a list of atmospheres that span all combinations of those values (similar to meshgrid).

Usage

MakeAtmList.lin(z0 = 0, c0 = 343, gc = 0, wx0 = 0, gwx = 0, wy0 = 0, gwy = 0, rho0 = 1.2929 * exp(-z0/68))

Arguments

- z0: Elevation of intercept layer (m)
- c0: Intrinsic sound speed at intercept layer (m/s)
- wx0: Zonal (east-west) wind at intercept layer (m/s)
- wy0: Meridional (north-south) wind at intercept layer (m/s)
- rho0: Density at intercept layer (kg/m^3)
- gc: Vertical intrinsic sound speed gradient (1/s)
- gwx: Vertical zonal wind gradient (1/s)
- gwy: Vertical meridional wind gradient (1/s)
- grho: Vertical density gradient (kg/m^4)

Details

This is useful for modeling wave propagation in a range of atmospheres when actual atmospheric characteristics are poorly constrained.

Value

List of linear atmospheres.

Author(s)

Jake Anderson

Examples

# make atmospheres spanning a range of base sound speeds and
# sound speed gradients

c0 = seq(330, 336, 0.1)
c0 = seq(-0.006, -0.004, 0.0005)

ATM_list = MakeAtmList.lin(c0 = c0, gc = gc)
**meshgrid**

*Create a mesh grid like in Matlab*

---

**Description**

Creates 2D matrices for accessing images and 2D matrices

**Usage**

```plaintext
meshgrid(a, b)
```

**Arguments**

- `a`  x vector components
- `b`  y vector components

**Details**

returns outer product of x-components and y-components for use as index arrays

**Value**

- `x` length(y) by length(x) matrix of x indicies
- `y` length(y) by length(x) matrix of y indicies

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**Examples**

```plaintext
meshgrid(1:5, 1:3)
```
**meshgridn**

*Multi-Dimensional Meshgrid*

**Description**

Inputs a list of vectors, and returns a list of vectors such that every possible combination of input vector values occurs once in the output.

**Usage**

`meshgridn(L)`

**Arguments**

L  
list of vectors

**Value**

List of vectors of equal length such that each combination of elements drawn from the input vectors occurs exactly once in the output list.

**Author(s)**

Jake Anderson

**Examples**

`meshgridn(list(1:UL 1P:1RL 1PP:1PT))`

---

**P4X.lin**

*Find Ray Parameter for Distance*

**Description**

Calculates ray parameter and azimuth of a ray that reaches some given point, using a 2-D iterative optimization search.

**Usage**

`P4X.lin(x, y, zs, zr, ATM = CheckAtm.lin(list()), maxerror = 3)`
Arguments

- **x**: Abscissa of receiver relative to source (m)
- **y**: Ordinate of receiver relative to source (m)
- **zs**: Elevation of source (m)
- **zr**: Elevation of receiver (m)
- **atm**: Linear atmosphere
- **maxerror**: Maximum permitted error for solution (m)

Details

'maxerror' should be set with the necessary arrival time precision in mind. Smaller values of maxerror will produce more accurate arrival locations (and arrival times), but will require longer calculation times.

Value

List with following elements:

- **p**: ray parameter
- **az**: azimuth
- **error**: distance between inputs (x,y) and final position of result ray

Author(s)

Jake Anderson

Examples

```
ATM = CheckAtm.lin()
P4X.lin(100, 100, 100, 00, ATM)
P4X.lin(100, 100, 100, 00, ATM, 0.01)
```

Description

Calculates ray parameter and azimuth of a ray that reaches some given point, using a 2-D iterative optimization search. Does not use compiled C code, so is slower, but in case of some bug in the C code, this function will still work.

Usage

```
P4X.lin_no_c(x, y, zs, zr, ATM = CheckAtm.lin(), maxerror = 3)
```
Arguments

- **x**: Abscissa of receiver relative to source (m)
- **y**: Ordinate of receiver relative to source (m)
- **zs**: Elevation of source (m)
- **zr**: Elevation of receiver (m)
- **ATM**: Linear atmosphere
- **maxerror**: Maximum permitted error for solution (m)

Details

'maxerror' should be set with the necessary arrival time precision in mind. Smaller values of maxerror will produce more accurate arrival locations (and arrival times), but will require longer calculation times.

Value

List with following elements:

- **p**: ray parameter
- **az**: azimuth
- **error**: distance between inputs (x,y) and final position of result ray

Author(s)

Jake Anderson

Examples

```
ATM = CheckAtm.lin()
P4X.lin_no_c(100, 100, 100, 00, ATM)
P4X.lin_no_c(100, 100, 100, 00, ATM, 0.01)
```

Description

Makes a plot showing effective sound speed structure.

Usage

```
PlotAtm.lin(ATM, zlim = c(0, 100), winddir = 90, col = sky.colors(500),
TOPO = NULL)
```
Arguments

- **ATM**: Linear Atmosphere
- **zlim**: Height limits in plot
- **winddir**: Direction of wind to plot
- **col**: color vector, such as the output of heat.colors
- **TOPO**: list containing vectors x and y, and matrix z, or NULL

Details

When plotting topography, an east-west cross-section where TOPO$y$ is zero is plotted. TOPO should be formatted as a potential input to functions like `contour` or `image`.

Value

None; plot side effects only.

Author(s)

Jake Anderson

Examples

```r
ATM = checkatmNlin()
plotatmNlin(ATM)
```

---

**Prop.lin**  
*Acoustic Ray Tracing*

Description

Given a ray parameter, azimuth, source/receiver elevations, calculates where ray lands at receiver elevation.

Usage

```r
Prop.lin(p, az, zs, zr, ATM = CheckAtm.lin(list()))
```

Arguments

- **p**: ray parameter (s/m)
- **az**: azimuth (degrees)
- **zs**: source elevation (m)
- **zr**: receiver elevation (m)
- **ATM**: Linear atmosphere
Value

List including the following elements:

- \( x, y \) ending position of ray
- \( t \) arrival time of ray
- \( A \) arrival amplitude
- \( p \) ray parameter

Author(s)

Jake Anderson

References


Examples

```r
ATM = CheckAtm.lin()
prop.lin(0.001, 45, 100, 0, ATM)
```

Description

Given a ray parameter, azimuth, source/receiver elevations, calculates where ray lands at receiver elevation. Does not use compiled C code, so is slower, but in case of some bug in the C code, this function will still work.

Usage

```r
prop.lin_no_c(p, az, zs, zr, ATM = CheckAtm.lin(list()))
```

Arguments

- \( p \) ray parameter (s/m)
- \( az \) azimuth (degrees)
- \( zs \) source elevation (m)
- \( zr \) receiver elevation (m)
- \( ATM \) Linear atmosphere
Rescale a vector to fit in a certain range

Rescale a vector to fit in a certain range

RESCALE(x, nx1, nx2, minx, maxx)

Arguments

x vector
nx1 new minimum
nx2 new maximum
minx old min
maxx old max

Details

Used for graphics.
Value
scale vector is returned

Author(s)
Jonathan M. Lees<jonathan.lees.edu>

Examples
\[ x = \text{rnorm}(10) \]
\[ \text{RESCALE}(x, 3, 9, \text{min}(x), \text{max}(x)) \]

sky.colors

Sky Colormap

Description
Returns a vector of colors one would see in a sunrise, ranging from orange to blue.

Usage
sky.colors(n)

Arguments
n length of output color vector

Value
Vector of colors of length n.

Author(s)
Jake Anderson

Examples
\[ \text{col} = \text{sky.colors}(12) \]
\[ \text{M} = \text{matrix}(1:12, 12, 1) \]
\[ \text{image}(M, \text{col} = \text{col}) \]
Index

*Topic misc
  A2P.lin, 2
  CheckAtm.lin, 3
  MakeArrivals.lin, 4
  MakeAtmList.lin, 5
  meshgrid, 6
  meshgridn, 7
  P4X.lin, 7
  P4X.lin_no_c, 8
  PlotAtm.lin, 9
  Prop.lin, 10
  Prop.lin_no_c, 11
  RESCALE, 12
  sky.colors, 13

A2P.lin, 2

CheckAtm.lin, 3

MakeArrivals.lin, 4
MakeAtmList.lin, 5
meshgrid, 6
meshgridn, 7

P2A.lin (A2P.lin), 2
P4X.lin, 7
P4X.lin_no_c, 8
PlotAtm.lin, 9
Prop.lin, 10
Prop.lin_no_c, 11

RESCALE, 12

sky.colors, 13