User's Manual for the program *D2tracerdn*

Misha Barmin

Department of Physics
University of Colorado at Boulder

1. Overview

The primary purpose of this software is to compute Source Specific Station Corrections (SSSCs) in a 3D seismic velocity model relative to a 1D reference seismic velocity model. Alternatively, this code can generate travel time tables for a 3D model. Both the 1D reference model and the 3D model are provided by the user.

An SSSC for a given seismic phase at a given station is defined as the difference between travel times calculated in the 3D model and in the reference 1D spherical model (IASP91):

$$SSSC(\theta, \phi, h) = t^{3D}(\theta, \phi, h; \theta_0, \phi_0) - t^{1D}(\Delta, h) \tag{1}$$

where θ, ϕ, and h are the source latitude, longitude, and depth, θ_0 and ϕ_0 are station coordinates and Δ is the epicentral distance.

This software was developed at the CU (Boulder) Physics Department, and follows the format and conventions of that model. The definitions of seismic phases and the reference 1D model are external to the code, and may be changed by the user. In this manual, the IASPEI91 1D velocity structure is used as a reference in the computation of SSSCs, and the phase nomenclature follows that of the IASP91 travel time computation package.

Travel times (or SSSCs) from the 3D velocity model are computed using a 2D raytracing algorithm. Computations are performed for user-specified phases along a set of profiles radiating from the specified point. The 3D velocity field of the input model is interpolated along each profile. Subsequently, minimum travel times for each phase are estimated throughout the user-defined volume.

All I/O operations are performed in ASCII format. The **GMT** graphics package is used to visualize the results.

This manual covers the basic steps of the computation, the formats of relevant files, and provides examples of using the software. Some of the specifics of the algorithm are described in Appendix C.

2. Contents of the Distribution

The software package consists of the following elements:

- **D2tracerdn** — main code that performs ray tracing and computes travel times and SSSCs.
- **conv_cu_idc** – introduces elevation-related travel-time corrections into the SSSC file to accomodate their use in IDC software, and converts station-specific polar coordinates of the SSSCs

onto a geographic (lat,lon, value, error) grid.

- **iasp91_tbl** – prepares a look-up table of 1D reference model travel times taking into account specific parameters (distances, source depths etc.) specified by the user.
- **pr_sssc** – extracts SSSC values for one depth from the SSSC table produced by D2tracerdn and converts them into a three-column file for plotting (e.g., with GMT).
- **plotP_sssc, plotP_rad_ll, CC_plot** - plotting scripts utilizing the GMT package.

3. Computation Flowchart

3.1. Preparation:

3.1.1. . Obtain or assemble a 3D velocity model (see section 7 for format description)

3.1.2. . Assemble a look-up table of phase-specific travel-times and elevation corrections (see section 8 for format description and instructions)

3.1.3. . Compose an input parameter file for the computation (see section 4 and Appendix B)

3.2. Computation:

User Call: D2tracerdn parameter_file > logfile.

The code will perform the following sequence of operations:

3.2.1. . Read in the 3D model and run parameters (station location, distance and azimuthal step for ray-tracing, phases etc.)

3.2.2. . **MAIN LOOP**

3.2.2.1. : For each azimuth, average the 3D velocity field along the profile, perform an Earth flattening transformation, construct interfaces and velocity grids for ray-tracing.

3.2.2.2. : For each phase defined in the parameter file, compute the ray field in the plane of the profile.

3.2.2.3. : For each phase defined in the parameter file, form a table of first-arriving times in the plane of the profile, and compute SSSCs (if requested in the parameter file).

3.2.3. . Output tables with travel times or SSSCs, and tables of ray geometries (optional).

3.3. Post-processing and Visualization:

3.3.1. . Use shellscripts of *plot_sssc* family to make maps of the resulting SSSC field.

3.3.2. . Use shellscripts *CC_plot* to make images of ray geometries and velocity model values.

3.3.3. . Use code *conv_cu_idc* to "remove" elevation corrections and to convert from the radial (angle from station, distance, value) into geographic (lat,lon, value) grid (see section 6.1).

4. Runtime Parameters for D2tracerdn

The D2tracerdn code requires numerous run-time parameters and settings. An efficient way to feed these into the code is via a parameter file read at the begining of the run. Parameters are defined below, and an example of the parameter file is provided in the Appendix B. If the user intends to utilize the $IASP91$ 1D reference model and a CU (Boulder) 3D model (provided with the distribution), adapting the station-specific parts and the plotting option keys within the parameter file shown in the appendix (and also provided with the distribution) should be sufficient.

m_name - model name, any string up to 8 charaters, will be used to form the output file name.

nlayers - defines the number of layers in the input model. The number of layers should be constant across the model. Default value: 8.

st_rk - main integration step for the Runge-Kutta method. Default value: 0.2

wavet - Wave type P or S. Default value: P

grid_size - size of the grid cell for in the model, in degrees. Default value: 2

debug - If debug=1, additional info will be output on stdout and temporary files will not be deleted upon completion of the run; if debug=0 - no additional debug info is provided. Default value: 0

graph - Output auxiliary files with graphic information for plotting ray diagrams. Possible values are 0,1,2, where:
0 - will not produce any files with graphic info; 1 - will make files with diving waves; 2 - will make files with diving waves and direct waves. Default value: 0

NOTE: If using ray diagram output option, the output destination directory should contain a subdirectory called "GRAPH" where all files related to ray diagrams will be placed. See section 5.2 for details.

sta_name - Station name, up to 4 symbols.

sta_nick_name - Station "nick name", may be used as a surrogate name.

sta_lat, sta_lon, sta_elev -coordinates and elevation of the station.

start_az, end_az - Range of azimuths (from north, in degrees) for which ray tracing will be done. Default values: 0 and 357, respectively, yield a full circle. Give the same value for both to get only one azimuth.

step_az Increment in azimuth, in degrees, should be an integer. Default value: 3

The following block of parameters describes which rays will be traced in which parts of the model. See section 4.1 for a discussion of the overall strategy. Parameters are defined as follows: *#layer* - layer id number; *group* - group number; *SSSC* - suffix for the SSSC's filename for P and S wave type; *wave* - compute direct and diving waves; *head* - compute head waves; *Comments* - any comments, any text after # is ignored.

See ray diagram in Figure 1 for an illustration of the ray types defined below.

rays =
BEGIN

```
#layer  group   SSSC   wave  head  Comments
   1      1    Pg Lg    N     N    # soft sediments
   2      1    Pg Lg    N     N    # hard sediments
   3      1    Pg Lg    Y     Y    # upper crust
   4      1    Pg Lg    Y     Y    # middle crust
   5      1    Pg Lg    N     N    # lower crust
   6      2    Pn Sn    Y     Y    # upper mantle from Moho to 400 km
   7      3     P S     Y     N    # transition zone (400-670 km)
   8      3     P S     Y     N    # lower mantle (670 - 2740 km)
   END
```

NOTE: The code will use the first or second field in the SSSC definition field depending on the value of *wavet* parameter (P or S).

tables - compute SSSC or TTT tables. Default value: SSSC

delta - SSSC/TTT will be computed up to this distance (in degrees) from the station. The value should be between 3 - 22 degrees.

model_cu - filename of the 3D model, e.g. *../model/CUB2.0.raytracer.f.asc* (file provided with the release).

model_iasp - filename of the 1D reference look-up table, e.g. *./IASP91_TBL* (file provided with the release).

NOTE: This table is generated by the code *iasp91_tbl* that will take as input the same parameter file as the main code. The reference look-up file may be re-used for any station as long as the number of source depths and the maximum raytracing distance has not changed. See section 8 for general discussion and the format description.

out_dir - output destination directory for SSSC/TTT tables.

h - an array of values for source depths (in km), to be used in the computation of the SSSC/TTT tables. The maximum number of depths is 10. Depths must be listed in ascending order, and separated by spaces. If the number of depths is greater than 10, the first 10 are used. The maximum allowed value is 200 km.

4.1. Notes on the use of layer groups

Layer groups provide a way to define a desired phase; e.g. Pg, as a collection of rays that are contained within that group. In this way, travel times of rays traveling along complex paths through a 2D slice of the 3D model may be related to the travel times of phases in the 1D model. Care should be taken by the user to ensure that definitions used in the parameter file correspond to conventions used in constructing the 1D reference lookup table (in terms of penetration depths and propagation distances). *Example:* The definition of the Pn phase includes all rays that turn within layer 6, from Moho to 400 km (diving rays), as well as rays that travel along the **top** of that layer just below the Moho (head wave rays).

The definition of seismic phases also depends on the 3D model. For example, the CUB1.0 and CUB2.0 models are constructed from surface-wave data and, in some regions, the crust-mantle transition is not well constrained. As a consequence, the third crustal layer (lower crust) may not represent the typical crustal velocities. Therefore, we recommend excluding this layer from the definition of crustal phases Pg and Lg. However, the definition of crustal phases can be different for other models.

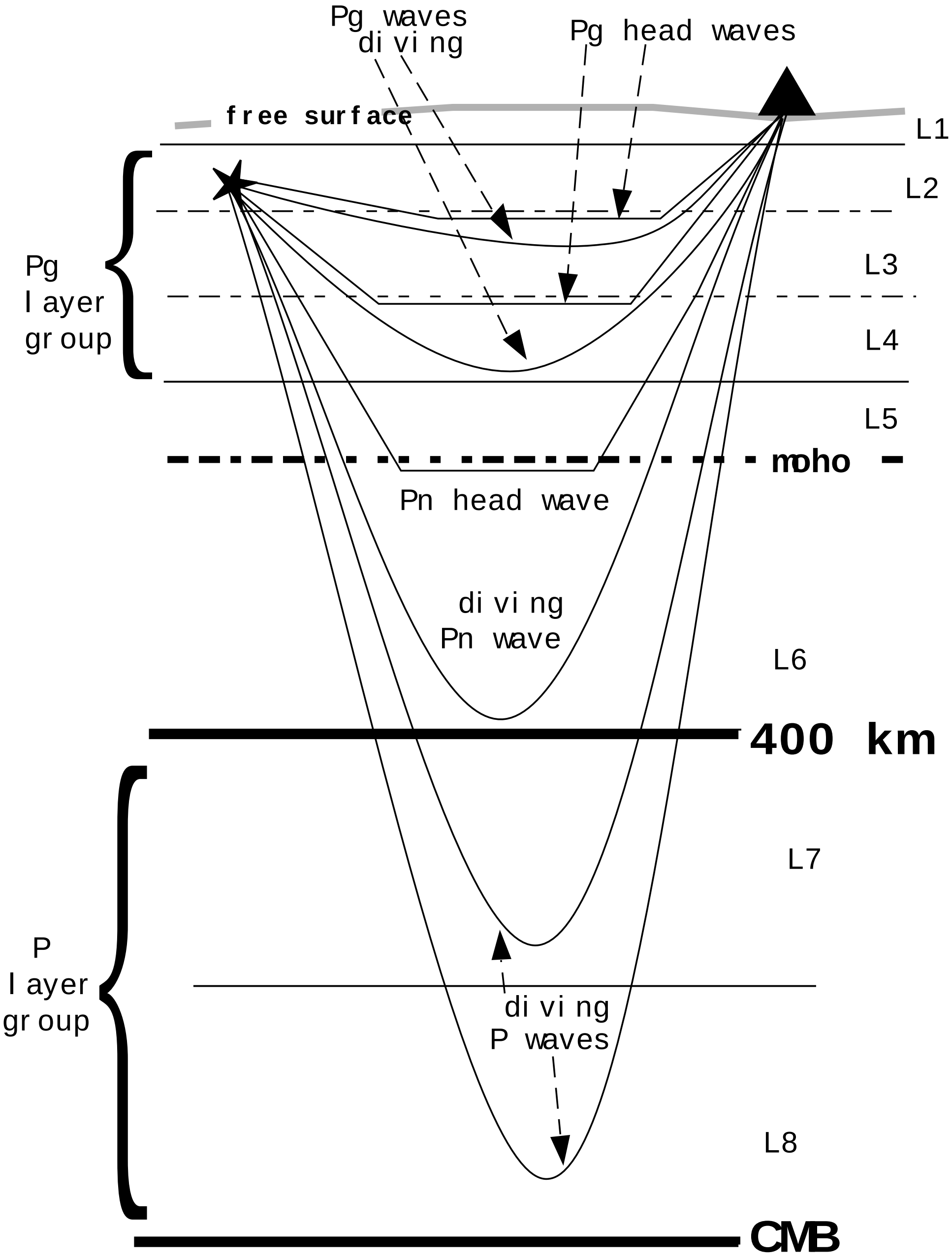


Figure 1. A schematic illustration of the layers, groups and rays defined in the example of the parameter file discussed in section 3. The file (shown in its entirety in Appendix B) is included in the release.

5. Output Files

5.1. SSSC tables

The primary product of the computation is the SSSC table for each of the phases requested. These tables are are placed into files with names like *EIL.Pg.CUB20.dat*, with station name (EIL), phase designation (Pg), and model name (CUB20) forming the filename.

The output table contains entries for each of the azimuths. Each entry is structured as follows:

The first line describes ray direction, station position, the number of source depths ns, the number of values n in the table, station name and nickname, and the phase designation. The second line provides depths of sources for which SSSCs will be computed. The subsequent n lines have the following fields:
first column - distance from the station, in km
second column - surface elevation taken from the 3D model, in km
subsequent ns pairs of values are (*travel time (sec); maximum ray depth (km))*.

Example: SSSC values for Pn phase computed along the profile towards the north (azimuth=0) from the seismic station EIL for 6 different depths. The overall length of the profile is 2475 km (99 increments of 25 km). Only a part of the file is shown below.

```
0.00000 29.66990 34.95120 0.20000 6 99 EIL EIL Pn
0.0 10.000 20.000 30.000 40.000 50.000
0.0 -0.655 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 -0.234 40.000 -0.227 50.000
25.0 -0.649 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 -0.337 40.000 -0.288 50.000
50.0 -0.613 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 -1.303 30.000 -0.545 40.000 -0.422 50.000
75.0 -0.540 9999.999 9999.999 -2.970 10.000 -1.949 20.000 -1.287 30.002 -0.682 40.000 -0.544 50.000
...
2475.0 -0.178 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999
```

NOTE 1. The value 9999.999 means that the phase is not computed for that combination of the source depth and distance.

NOTE 2. The *D2tracerdn* program evaluates travel times along rays computed in an elliptical Earth with the surface topography at the station taken from the 3-D model. Correspondingly, the values placed into the SSSC tables are relative to this station elevation. Because the IDC software will automatically correct for elevation, an "inverse correction" of the SSSC tables is required prior to use at the IDC. This correction is performed by the program *conv_cu_idc* provided in the package, see section 6.1.

5.2. Ray Geometry files

Files generated when parameter *graph* is non-zero are placed into a main output directory's subdirectory called *GRAPH*. File formats are for use with GMT plotting commands; e.g. the script **CC_plot** provided with the distribution.

File *EIL.P.fld.345.xyz* contains a description of the P wave velocity distribution along the profile with $azimuth = 345°$, in the form of a 3-column table (distance, depth, value).

File *EIL.Pg.ray.345.xy* contains a description of the ray geometry for rays of the Pg phase, in the form of a multisegmented 2-column table (distance, depth).

6. Supplementary Software

6.1. conv_cu_idc

The program is invoked with the following call:

conv_cu_idc station_name phase elevation depth in_dir iasp91_tbl_file model_name output_file

where station elevation is in km (positive down), *depth* is the source depth in km, *in_dir* is the directory holding the raw SSSC table, *model_name* is a string designating the model, and *iasp91_tbl_file* is the 1D lookup table file. See section 8 for the description of the lookup table file. The output of this code will be a four-column table, with fields:
latitude, longitude, SSSC value, SSSC error.
The error value is presently set to 0.

Travel times calculated in the 3D model, t^{3D}, include the effect of the station elevation. However, the model topography can be different from the real topography. For example, the topography of the CUB model has been smoothed because the model is defined on a $2°x2°$ grid. As a consequence, at some points, the difference between real and model elevations may be significant. To avoid this discrepancy, the shell **run_conv_cu_idc** reads the model elevation from the input SSSC table in CU format, and passes this value as an input parameter **elevation** for **conv_cu_idc** (see section 9.3). The resulting SSSC corresponds to the station located at the sea level:

$$SSSC^{s.l.} = SSSC - \Delta t_{elev} \tag{2}$$

6.2. pr_sssc

This program is invoked with the following call:

pr_sssc sta_name phase depth sssc_dir model_name

where *depth* is the source depth in km, *sssc_dir* is the directory holding SSSC tables, *model_name* is a string designating the model, same as was used in the parameter file (forms a part of the SSSC table file name).

This program reads the SSSC table and creates a 3-column table of SSSC values interpolated on a geographic grid (lat,lon, value). Output is intended for use in plotting; e.g., with GMT.

6.3. iasp91_tbl

This program computes reference 1D travel time tables for the raytracer **D2tracerdn** at specified depths and in a specified distance range. It is invoked with the following call:

```
iasp91_tbl parameter_file
```

iasp91_tbl uses the same **parameter_file** as **D2tracerdn** and requires an environment variable **IASPEI91_TABLE** to be defined and to indicate the location of the IASPEI tables. To change the depths and the distance range for SSSCs the following steps are required:

1. Edit parameter file
2. run iasp91_tbl (see macro below)
3. run D2tracerdn

```
———— macro to run iasp91_tbl (from EIL_test/ directory)
#!/bin/csh
#
# Create IASP91_TBL table
#
setenv IASPEI91_TABLE ”../src/iasp_tbl/iasp91”
../bin/iasp91_tbl par_file_P_EIL
exit
```

7. 3D velocity model

7.1. General description

The model is defined on a partially irregular 3D grid in a geographic spherical coordinate system. It is regular in latitude and longitude (either a 1°x1° or a 2°x2° grid), and is irregular in depth. The model at each geographical point is defined by a 1D profile of P and S velocities. Each 1D profile consists of a set of *knots*. A knot is a node at a specific depth with P and S velocities specified. The number of knots from the CMB to the surface must be constant at each geographical point throughout the model.

Interfaces and layers are defined in the following way. Each layer in the model is delimited by two knots with the same value of depth. The number of knots that constitute each layer must be the same across model. However, the number can be chosen by the user as long as it is over 2 and under 48, and the total number of knots from the CMB to the surface is less than 100. The number of knots defining individual layers is arbitrary. For example, 5 knots may define layer n and 7 may define layer $(n+1)$. However, these values must be constant throughout the model.

Not all layers defined at each geographical node need to be realistic. To satisfy uniformity of layer definitions throughout the 3D model the user may find it helpful to introduce ”fictitious layers”. A fictitious layer may be represented by a few knots (at least 2), with small depths difference, e.g. $\varepsilon = 1$m. This could be useful to ensure that the number of knots in each vertical profile is geographically constant. The velocities in the fictitious layer should not differ significantly from the velocities in the adjoining layers.

7.2. Model definition format

The model is defined on either a 2°x2° or a 1°x1° geographical grid. At each geographical node it represents a 1D vertical profile of P and S velocities. Each profile is described by a set of knots and the node's position. Each node is described by a sequence of lines, formatted as follows. A one-line header with 3 free-format fields: lat, lon, n:

lat - geographical latitude, (deg), ($real$),

lon - geographical longitude, (deg), ($real$),

n - number of vertical knots, ($integer$).

Following the header is a table containing n rows (knots). Each knot is described by 5 fields: $id, kinfo, H, Vp, Vs$, (format - free):

id - knot id number, $k = 1, ..., n$, ($integer$),

$kinfo$ - not used,(some information about the knot, reserved for further usage), ($integer$),

H - knot depth, (km), ($real$),

Vp - P-wave velocity (km/s), ($real$),

Vs - S-wave velocity (km/s), ($real$).

Individual nodes may be arranged in an arbitrary way within the file. No empty lines should be present within the file. Layers at the top and the bottom of the model are defined by a single knot each.

See Appendix A for an example of the vertical velocity profile.

8. 1D reference model and the travel time look-up table

The look-up table of travel times and elevation corrections for the phases used in the computations is constructed on the basis of a 1D model by the program $iasp91\_tbl$.

It is based on the **iasp91** package, and uses the vertical profile of seismic velocities provided therein. If desired, it is possible to substitute a different velocity profile within the **iasp91** package. Alternatively, the user may change the 1D reference model altogether, along with the nomenclature of seismic phases used. In this case the computation of the 1D reference look-up table will be up to the user.

Travel times within this table are used to compute SSSC values. Elevation corrections may be applied to the SSSCs. This operation, performed by the program *conv_cu_idc*, is intended as a counter-action in anticipation of the mandatory elevation correction within the CMR software, and should yield "true" values in the operationally used SSSCs.

The look-up table file contains entries for each of the phases used, in sequence. Each entry is structured as follows:

The first line describes the ray direction, the station position (these are always set to 0 as the computation is done in polar coordinates, and does not depend on direction), the number ns of source depths, the number n of values in the lookup table, two junk strings, and the phase name. The second line provides source depth values for which SSSCs will be computed. The subsequent n lines have the following fields:
first value - distance from the station, in km
subsequent ns values - travel times, sec
subsequent ns values - elevation corrections, in *sec/(km of elevation)*

Example: look-up reference table computed for the phase Pg and 6 source depths out to a distance of 2475 km (99 increments of 25 km).

```
0.00000 0.00000 0.00000 0.00000 6 99 IASP IASP Pg
0.0 10.000 20.000 30.000 40.000 50.000
0.0 0.001 1.724 3.448 4.987 9999.999 9999.999 0.000 0.172 0.172 0.172 9999.999 9999.999
25.0 4.310 4.645 5.504 6.482 9999.999 9999.999 0.000 0.065 0.110 0.136 9999.999 9999.999
50.0 8.621 8.791 9.234 9.625 9999.999 9999.999 0.001 0.032 0.079 0.101 9999.999 9999.999
75.0 12.931 13.044 13.068 13.272 9999.999 9999.999 0.001 0.025 0.079 0.087 9999.999 9999.999
100.0 17.241 17.318 16.902 17.026 9999.999 9999.999 0.001 0.020 0.079 0.083 9999.999 9999.999
125.0 21.551 21.517 20.736 20.816 9999.999 9999.999 0.002 0.079 0.079 0.081 9999.999 9999.999
....
2475.0 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999 9999.999
```

NOTE: The value 9999.999 means that the phase is not computed for that combination of the source depth and distance.

9. Practical Example

Task Use the CUB2.0 model to compute SSSCs for compressional phases (P, Pn, Pg) at seismic station EIL (Eilat, Israel). Generate maps of SSSCs and a diagram of rays along an azimuth of 345° from the station.

The master parameter file, named *par_file_P_EIL* required for this task is in Appendix B. All scripts for plotting, and relevant color tables, are in the *PLOTTING-SCRIPTS* subdirectory of the release.

The sequence of actions is as follows:

9.1.

Running the ray tracer.

The directory where the run is performed should contain the 1D reference look-up table named *IASP91_TBL*. The file with a 3D model is expected to reside in *../model/CUB2.0.raytracer.f.asc*. Capture the screen output into a logfile if desired.

Since ray diagrams are very bulky, it is advisable to compute the diagram only along the direction of interest. Change run parameters **start_az, end_az** to be of the same values, in this case 345. A subdirectory *GRAPH* must exist.

Issue this command:

```
../bin/D2tracerdn par_file_P_EIL_az345rays > logfile
```

This will populate the GRAPH directory with ray-geometry files and velocity profile files for a slice along 345° from the station. See section 6.2 for name conventions and format descriptions.

Next perform a run for maps of SSSCs. This order of operation is necessary since files holding SSSC tables are overwritten during the run.

Issue this command:

```
../bin/D2tracerdn par_file_P_EIL > logfile
```

This will produce a set of 3 files in the current working directory, named *EIL.P.CUB20.dat, EIL.P.CUB20.dat, EIL.P.CUB20.dat.*

9.2.

Plot maps of all SSSCs for the source at 10 km depth using the model CUB1.0.

```
plotP_sccc EIL
```

This yields a PostScript file named *Map.EIL.P.sssc.ps*, shown on Figure 2.

9.3.

Transform SSSC values for Pn and Pg from station-specific polar coordinates to geographic (lat,lon) coordinates, and plot.

```
run_conv_cu_idc EIL
```

A set of files will be generated:

TT.EIL.Pg.reg.mideast, TT.EIL.Pn.reg.mideast are IDC-format SSSC tables. *sssc.EIL.Pg, sssc.EIL.Pn* are SSSC tables interpolated onto latitude-longitude grids.

```
plotP_rad_ll EIL
```

This yields a PostScript file named *Map1.EIL.P.sssc.ps*, shown on Figure 3 for the 3-D model CUB1.0.

9.4.

Plot ray geometry along the profile with $azimuth = 345°$.

```
cd GRAPH
CC_plot EIL 345 P
```

This yields a PostScript file named *GRAPH/EIL.P.345.ps*, shown on Figure 3 for the 3-D model CUB1.0.

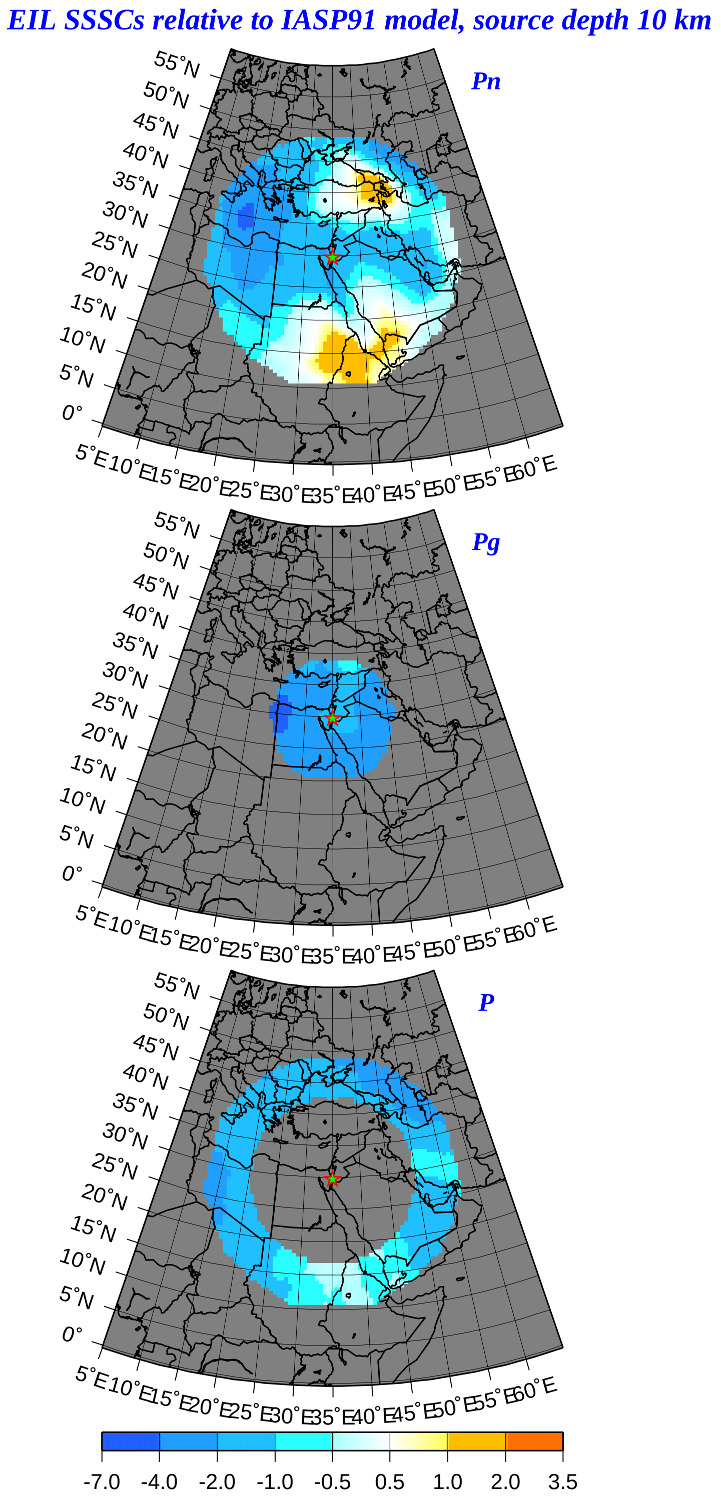


Figure 2. SSSCs computed at seismic station EIL (Eilat, Israel) for Pg, P_n and P_g wave groups in the CUB1.0 3-D model relative to the predictions of IASPEI91 1D Earth structure. Hypothetical source is at 10 km depth. Scale is in seconds.

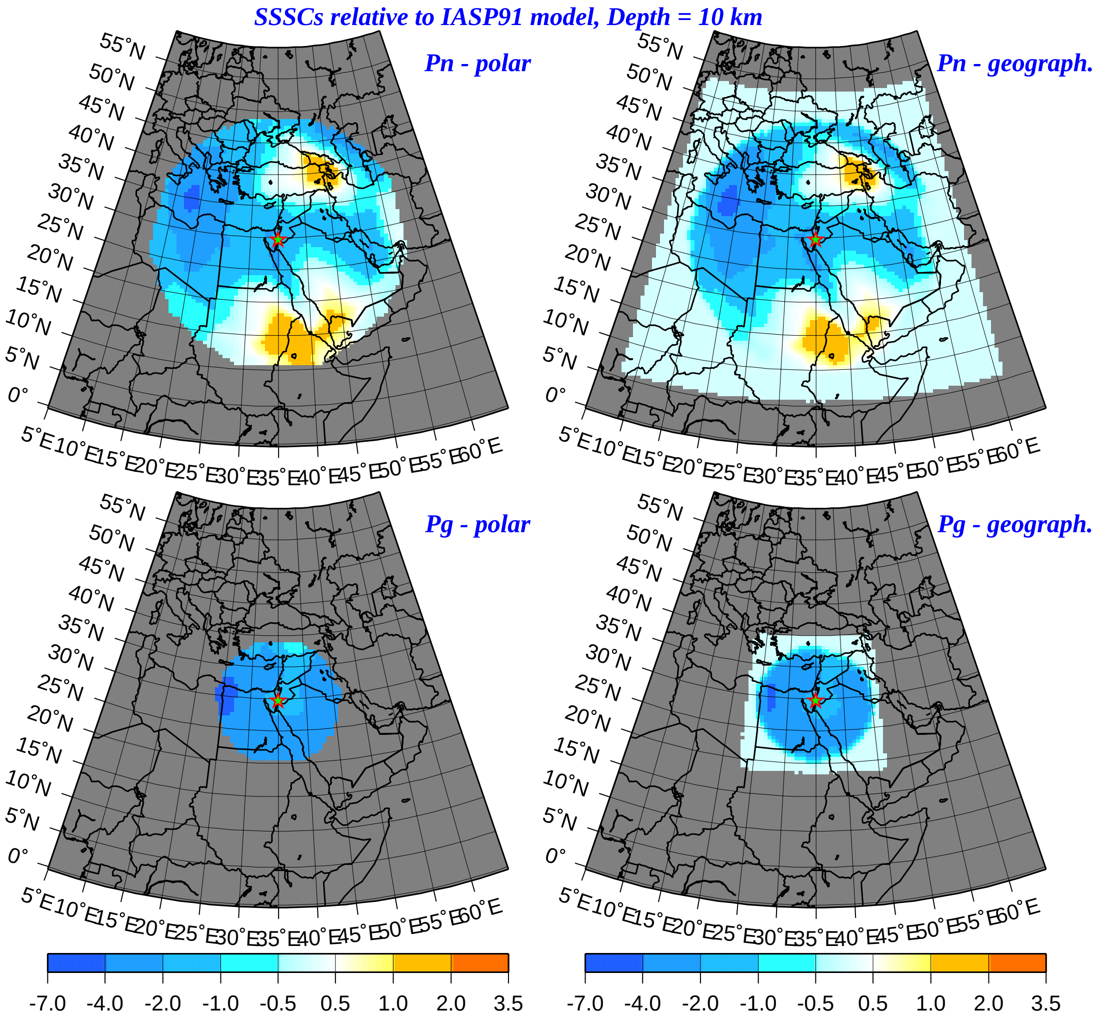


Figure 3. SSSCs computed at seismic station EIL (Eilat, Israel) for Pg and P_n wave groups in the CUB1.0 3-D model relative to the predictions of IASPEI91 1D Earth structure. The hypothetical source is at 10 km depth. Scale is in seconds. The left column shows a radial registration of SSSC values native to the computation, the right column shows the same values mapped into a geographic coordinate grid.

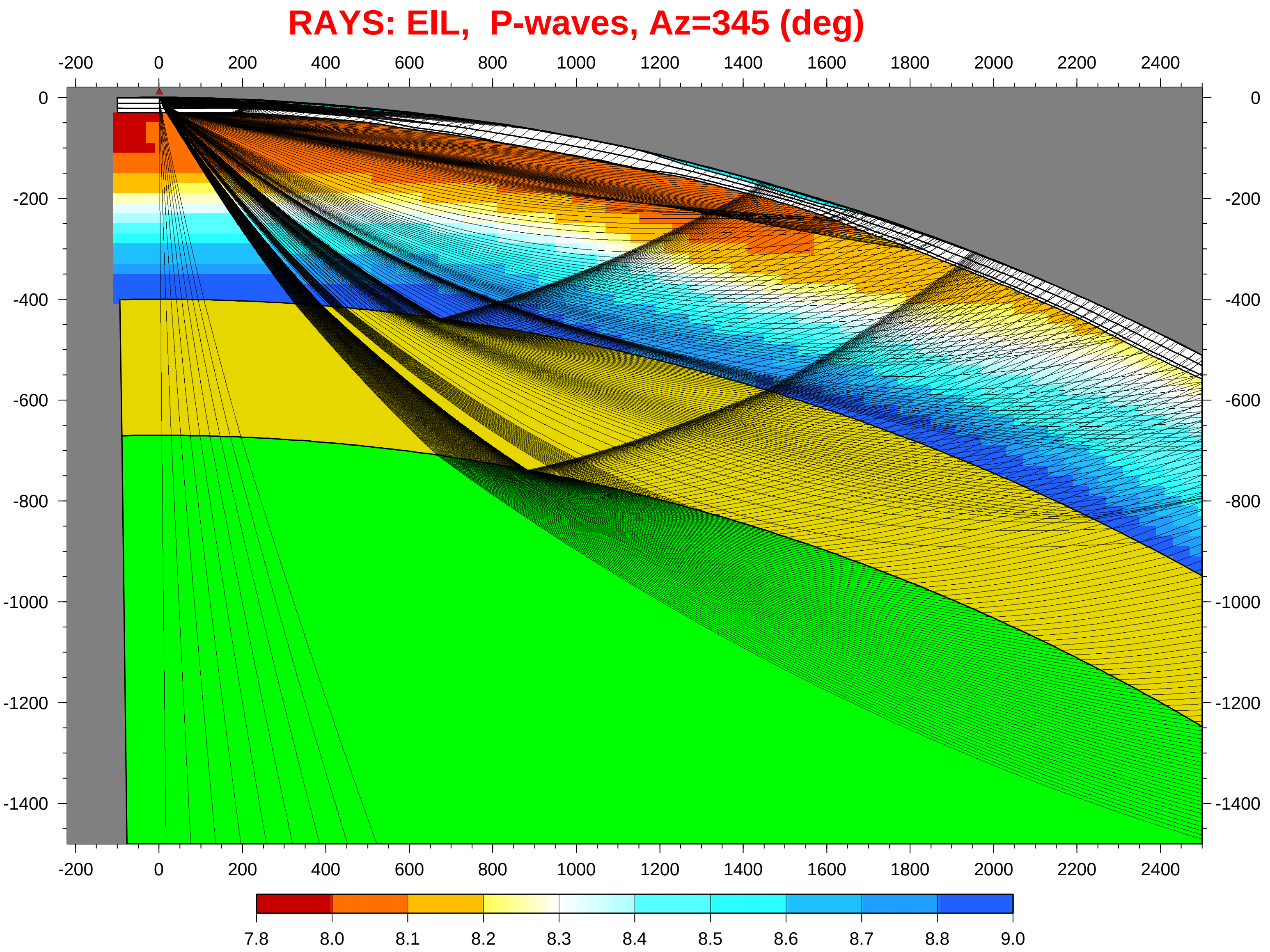


Figure 4. Ray geometry diagram for the azimuthal direction 345° from station EIL. Velocity field is interpolated from the CUB1.0 model. Color scale is in km/s, axes are labeled in km.

10. Appendix A

Vertical Velocity Profile Example

40.0000	68.0000	73			__ Free surface, interface #1
1	167	-0.4100	2.5000	1.1970	Layer #1
2	166	0.5900	2.5000	1.1970	__ Interface #2
3	165	0.5900	3.8662	1.9680	Fictitious layer #2
4	164	0.5901	3.8662	1.9680	__ Interface #3
5	163	0.5901	6.0978	3.3643	Layer #3
6	162	20.5210	6.0978	3.3643	__ Interface #4
7	161	20.5210	6.3636	3.7930	Layer #4
8	160	40.0925	6.3636	3.7930	__ Interface #5
9	159	40.0925	7.3607	4.0350	Layer #5
10	158	40.8550	7.3607	4.0350	__ Interface #6
11	157	40.8550	8.0156	4.4403	
12	156	41.3156	8.0174	4.4422	
13	155	41.7761	8.0191	4.4441	
14	154	42.2367	8.0209	4.4460	
15	153	42.6972	8.0227	4.4479	
16	152	43.1578	8.0244	4.4498	
17	151	43.6183	8.0262	4.4517	
18	150	44.0789	8.0280	4.4536	
19	149	44.5394	8.0297	4.4555	
20	148	45.0000	8.0315	4.4574	
21	147	50.0000	8.0475	4.4745	Layer #6
22	146	60.0000	8.0633	4.4914	
23	145	70.0000	8.1003	4.5310	
24	144	80.0000	8.1498	4.5838	
25	143	108.0000	8.1925	4.6426	
26	142	136.0000	8.2545	4.6698	
27	141	164.0000	8.3162	4.6582	
28	140	192.0000	8.3380	4.6047	
29	139	220.0000	8.3872	4.5887	
30	138	242.0000	8.4751	4.6366	
31	137	265.0000	8.5541	4.6729	
32	136	287.0000	8.6268	4.7032	
33	135	310.0000	8.7113	4.7438	
34	134	332.0000	8.7871	4.7774	
35	133	355.0000	8.8610	4.8070	
36	132	377.0000	8.9345	4.8383	
37	131	400.0000	8.9670	4.8716	__ Interface #7
38	131	400.0000	9.3706	5.0858	
39	129	433.3300	9.4715	5.1475	
40	128	466.6700	9.5807	5.2166	
41	127	500.0000	9.6599	5.2694	Layer #7
42	126	533.3300	9.7222	5.3129	
43	125	566.6700	9.7957	5.3625	
45	123	600.0000	9.9117	5.4353	
46	122	670.0000	10.2322	5.6277	__ Interface #8

47	121	670.0000	10.8241	5.9789	
48	120	710.0000	10.9229	6.0797	
49	119	760.0000	11.0558	6.2095	
50	118	809.5000	11.1440	6.2474	
51	117	859.0000	11.2300	6.2841	
52	116	908.5000	11.3140	6.3199	
53	115	958.0000	11.3960	6.3546	
54	114	1007.5000	11.4761	6.3883	
55	113	1057.0000	11.5543	6.4211	
56	112	1106.5000	11.6308	6.4530	
57	111	1156.0000	11.7055	6.4841	
58	110	1205.5000	11.7787	6.5143	
59	109	1255.0000	11.8504	6.5438	Layer #8
60	108	1403.5000	12.0568	6.6280	
61	107	1502.5000	12.1881	6.6809	
62	106	1601.5000	12.3151	6.7317	
63	105	1651.0000	12.3772	6.7564	
64	104	1700.5000	12.4383	6.7807	
65	103	1750.0000	12.4987	6.8046	
66	102	1898.5000	12.6759	6.8745	
67	101	1997.5000	12.7915	6.9199	
68	100	2146.0000	12.9625	6.9870	
69	99	2294.5000	13.1325	7.0540	
70	98	2393.5000	13.2462	7.0991	
71	97	2492.5000	13.3610	7.1449	
72	96	2542.0000	13.4189	7.1681	
73	95	2690.5000	13.5961	7.2398	
74	94	2740.0000	13.6564	7.2645	__ Bottom interface #9

The above table shows a vertical profile of velocities from the node at 40N, 68E within the CUB1.0 model. The total number of knots is 73. Layer #2 is ”fictitious”.

11. Appendix B

Using the parameter file below will result in computation of 3 SSSC tables (Pg, P_n, P waves) for the station EIL (Eilat, Israel) in the 3D velocity model CUB2.0.raytracer.f.asc.

```
# This is an example of parameter file for D2tracerdn v1.1
# program. This file is working with CUB1.0 model.
#
# Model name, max 8 character
#
m_name='CUB20_J362D28'
#
#
nlayers=8
#
# Parameter nlayers defines the number of layers in the input
# model. The number of layers should be constant across model
# otherwise program is terminated with error message
# (E001) Wrong number of layers .... Default value: 8
#
nlayers=8
#
# Main integration step by Runge-Kutta method. Default value: 0.2
#
st_rk=0.2d0 # seconds
#
#
# Wave type P or S. Default value: P
#
wavet=P
#
# Size of grid cell for model, (deg). Default value: 2
#
grid_size=2
#
# If debug=1 additional info ouput on stdout and temporary files
# will not be deleted upon completion of the run,
#if debug=0 - no additional debug info is provided.
# Default value: 0
#
debug=0
#
# Output auxiliary files with graphic information:
# graph=0/1/2, where
# Default value: 0
# 0 - do not produce any files with graphic info;
# 1 - diving waves;
# 2 - diving waves and direct waves.
# !!Make sure directory GRAPH exists as a subdirectory of the
# output directory set by parameter out_dir!!!
graph=0
#
# Station name, up to 4 symbols
#
sta_name=EIL
#
# Station nick name, may be used as surrogate name
#
sta_nick_name=EIL
```

```
#
# Station latitude, (deg)
#
sta_lat=29.6699
#
# Station longitude, (deg)
#
sta_lon=34.951199
#
# Station elevation, (km)
#
sta_elev=0.200
#
#Start azimuth, (deg), should be integer.
# Default value: 0
#
start_az=0
#
#End azimuth, (deg), should be integer.
# Default value: 357
#
end_az=357
#
#Step in azimuth,(deg), should be integer.
# Default value: 3
#
step_az=3
#
# Ray computation parameters
# The following table contains layers properties:
# #layer - layer id number,
# group - group number
# SSSC - suffix for the SSSC's filename for P and S wave type
# wave - computation direct and diving waves
# head - computation heading waves
# Comments - any comments, any text after # is ignored
#
rays =
BEGIN
# #layer group SSSC wave head Comments
1 1 Pg Lg N N # soft sediments
2 1 Pg Lg N N # hard sediments
3 1 Pg Lg Y N # upper crust
```

```
4 1 Pg Lg Y N # middle crust
5 1 Pg Lg N N # lower crust
6 2 Pn Sn Y Y # upper mantle from Moho to 400 km
7 3 P S Y N # transition zone (400-670 km)
8 3 P S Y N # lower mantle (670 - 2740 km)
END
#
# Compute SSSC or TTT. Default value: SSSC
#
tables=SSSC
#
# SSSC/TTT radius around station. Should be in range: 3 - 22 degree.
#
delta=22.0
#
# Path to input 3D model file
#
model_cu='../model/CUB2.0.raytracer.f.asc'
#
# Path to input reference file
#
model_iasp='./IASP91_TBL'
#
# Path to output directory for SSSC's
#
out_dir=.
# Parameter h defines depths. Maximum number of depths is 10.
# Depths must be arranged from the smallest to the largest.
# If #h greater than 10, the first 10 are used. Maximum value h = 200 km.
h = 0. 10. 20. 30. 40. 50.
```

12. Appendix C: Methodology and Algorithm

12.1. Model of the Medium

12.1.1. Interfaces, layers, parameters . The model M is defined in a 3-dimensional volume, V:

$$V : x^i_{min} \leq x^i \leq x^i_{max}, \quad \text{where } \; i = 1, 2, 3; \;\; (x^1, x^2, x^3) = (x, y, z),$$

by functions specifying the distribution of P- and S-wave velocities (v_p and v_s, respectively). The parameters of the medium must be smooth functions of the coordinates inside the model space, meaning that parameters and their first and second derivatives are continuous. Smoothness may be violated at a finite set of interfaces across which the parameters or their first or second partial derivatives may be discontinuous.

We consider models in which every interface may be covered by a finite number of smooth surfaces $\mathcal{L}: f(x^i) = 0$ with the following property: the functions $f(x^i)$ are defined and are smooth in the whole model space V, i.e. $f(x^i)$ and their first and second derivatives are continuous everywhere in $\mathcal{V}$. They traverse the entire volume V, and end at its boundaries. Each surface divides the whole space into two parts, the positive parts, F^+ in which $f(x^i) > 0$, and the negative parts, F^- in which $f(x^i) < 0$. Accordingly, the side of the surface that faces the positive part of the space is called the positive side of the surface $\mathcal{L}$, the other one being the negative side of the surface $\mathcal{L}$.

We construct the model from simple blocks formed by smooth surfaces $\mathcal{L}$, where each simple block represents the physical unit of the model - a layer.

A simple block is defined by two finite sets F^+ and F^- of surfaces $f(x^i) = 0$. A point x^i lies within the block if and only if

$$f(x^i) \geq 0 \text{ for any } f \in F^+$$

$$f(x^i) \leq 0 \text{ for any } f \in F^-$$

The simple block is an intersection of the positive part of the volume space corresponding to the surfaces from F^+ and negative part of the volume space corresponding to F^- (see Gjoystdal *et al., 1985*). Some parts of the surfaces $\mathcal{L}$ may be boundaries of a block , while the remaining parts are only fictitious extensions of the boundary.

For complete ray tracing we will use a model with a simple layered structure. The layers are separated by non-intersecting or partially coinciding smooth interfaces, Figure 5. Representation of this structure in terms of simple block is trivial. Each interface is represented by one smooth surface $\mathcal{L}$, and each simple block represents the physical unit of the model - a layer. As Figure 5 shows, parts of some interfaces may coincide, which implies zero thickness for the layer. This trick gives us the possibility to introduce isolated blocks with edges. For ray tracing, we will actually use a very small number, ε, to represent zero thickness.

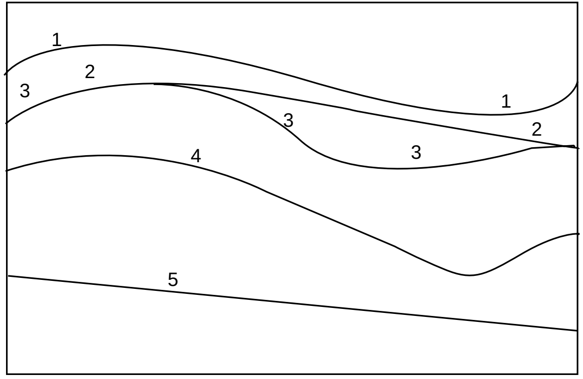


Figure 5.

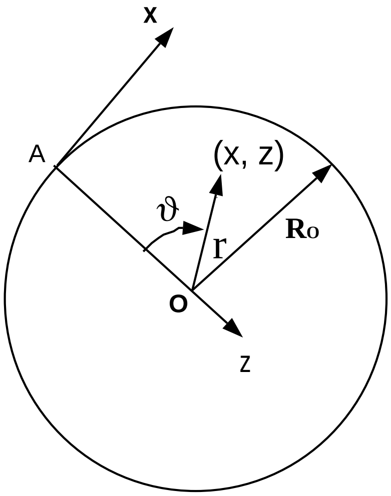
12.2. Transformations and Interpolation

D2tracerdn program performs two-dimensional (2D) ray tracing a three-dimensional (3D) model via an introduction of 2D profiles. Suppose that we fixed a basic point, A, on the free surface and have chosen an azimuthal direction. A 2D profile is an intersection of the 3D model with the great circle plane which passes through the point A along that fixed azimuth.

1. Reducing a 3D model to a 2D model.

   In the great circle plane we introduce geocentric polar coordinates $(x, z) = (\delta, H)$ as follows: δ is an angle between an axis OA and an arbitrary point (x, z) (positive direction along the azimuth), where O is the center of the earth. $H = R_0 - r$, where R_0 is the Earth's radius and r is the radial distance, see Figure 1. A new 2D profile in polar coordinates is defined in the following way. First chose a set of points at zero depth, and with a fixed step in δ (distance from the start of the profile, i.e. - from the station), and compute their positions in geocentric and, later, in geographic spherical coordinate system. Step values used in the code are 1° or 2°, depending on the size of cells the input 3D model. Since the 3D model is defined on a regular grid in geographical coordinates we use bilinear interpolation to get the depths and velocities for the set of points along the profile. This process leads to the creation of 1D vertical profile in 2D polar coordinates. The set of 1D vertical profiles forms a 2D grid (i.e., a 2D model), regular in δ and irregular in depth H.

   Note that the interfaces - layers policy described in section 2.1 is still valid. Instead of surfaces we now have curves.



2. Flattening the 2D polar model

   We introduce the next conformal transformation:

$$x = R_0\delta, \quad z = R_0 \ln \frac{R_0}{R_0 - H}, \quad v_{xz} = v_{polar} \exp \frac{z}{R_0},$$

   where, (x, z) are Cartesian coordinates, x is distance, y is depth, v_{xz} is the velocity in Cartesian coordinates, v_{polar} is the velocity in polar coordinates. This transformation preserves angles and travel times, therefore it does not matter in which coordinate system the following integral is computed:

$$t = \int_L \frac{\mathbf{ds}}{v},$$

where L is ray path. Using this operation we transform the crossection 2D profile of a 3D model into 2D Cartesian coordinates (distance, depth) which describe the model space for the 2D raytracer algorithm.

3. Constructing Interfaces.

   The final 2D model at this point is a regular grid in distance, x, and an irregular grid in depth, z. We also may extract From the input model we also obtain the information about interfaces. Each interface consists of the set of pairs, $(x_i, z_i), \quad i = 1, ..., n$, that define a function of one argument at a discrete set of distances. To expand this function along the whole range of arguments we use the fifth order polynomial interpolation of in each interval (x_i, x_{i+1}). This technique behaves similar to the spline interpolation, and provides continuos first and second derivatives in x everywhere, as well as a nonintersecting set of interfaces.

4. Velocity field interpolation.

   We use B-spline interpolation technique to obtain velocity values. For this purpose we have to create a file, regularly gridded in both distance and depth. Note that with the construction of interfaces we know layers boundaries.

   Two adjoining interfaces define a model layer with curvilinear boundaries. Our goal is to construct a rectangular block which covers the chosen layer. To do that we search for maximal and minimal values of depth and distance on the bounding interfaces. Using these values we construct a rectangular block that includes the layer, and make it a little bit (5 km) wide. Applying a linear extrapolation for the velocities and step regularization in depth we get a regular grid for the B-spline interpolation. The need for a wider block is dictated by the integration procedure, in cases when rays cross the interface. After constructing regular grids for all layers we may compute the velocity field at each point inside each layer.

12.3. Codes for Elementary Waves

12.3.1. Code definition. The D2tracerdn program performs raytracing for seismic body waves such as direct, refracted and reflected waves. Incidence of a wave at an interface of a simple block produces four waves, reflected P and S, and transmitted P and S. Converted waves are not considered during raytracing, and thus P and S waves are treated separately. At each incidence point we have to decide on which of the two resulting waves to follow. The numeric sequence specifying the behavior of the ray from its initial point to its end point is called its "code".

We apply a term "elementary wave" to the part of the wavefield that is described by the specific code. We also define a term *element of the ray*, which has an important meaning in the construction of the ray code. By an element we mean a part of the ray within one layer between two successive points of refraction/transmission, or between either the initial point or the endpoint of the ray and the closest point of reflection/transmission, or between the initial and the end point of the ray if the ray is entirely within one layer.

Here we describe possible types of codes of elementary waves as used in the D2tracerdn program. Let us assign integer numbers to interfaces from the free surface down to the bottom of the model, starting from one, and with a unit step. It is possible to use these numbers as layer numbers, i.e. the layer number is the number of its upper interface. If we consider any chosen ray, it will be divided into elements, each of them lying between two successive points where the ray strikes an interface. If the points of an element lie on different interfaces, it is called a *simple element.* When the end points lie on the same interface it is called a compound element. Any compound element is formally regarded as two simple elements. If we collect layer numbers along the ray we get the numerical code, for example, for a P wave,

$$1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 6\ \ 5\ \ 4\ \ 3\ \ 2\ \ 1,$$

and for S wave,

$$-1\ -2\ -3\ -4\ -5\ -6\ -6\ -5\ -4\ -3\ -2\ -1.$$

Positive numbers mean that all elementary waves are P waves, negative numbers mean that all elementary waves are S waves. D2tracerdn program uses the implementation of the ray tracer which does not support converted waves, so the numeric code has a constant sign. In an example above ray goes from the first layer through layers 2, 3, 4, 5 and it has a compound element in the sixth layer. This case corresponds to a diving wave which has a turning point in the sixth layer. After passing the turning point the ray goes through layers 5, 4, 3, 2, 1 up to the free surface. The same numerical code may be used to construct a primary reflected wave from the 7th interface. At present the raytracer does not use reflected waves completely, but only the part of the ray from the staring point to the reflection point. The program considers this part of a reflected wave as a direct wave. The starting point of the ray may be at any point of the model, including its boundaries. The source located on an interface (except the last interface) is considered as a point within the layer below that interface. The end point of a ray should be located either on the interface or on the free surface.

The above logic is applicable for all direct, diving and reflected rays, but does not apply to rays of refracted waves.

This is a special type of waves and corresponding rays. Consider an example of a refracted wave on the crust-mantle boundary. At first the ray goes from the free surface, and strikes the interface corresponding to the Moho boundary at a critical angle. It then goes along the Moho in small (5 km) steps. After each step the ray attempts to enter the layer under the the Moho along the direction of a tangent to the Moho interfac. In case of success the ray goes some distance in that layer, then strikes the Moho interface again, and goes up towards the free surface. We introduce this treatment of “refracted” waves as some approximation of the true refracted waves in order to avoid shadow zones which arise with changing Moho depth. Another possibility, when a ray takes off at a critical angle towards the free surface, is not considered in this program.

12.4. Theory. Runge-Kutta Method

Let us consider the main differential equation for solving the ray tracing problem:

$$\begin{aligned} \frac{dx}{dt} &= pv^2, \\ \frac{dz}{dt} &= qv^2, \\ \frac{dp}{dt} &= -\frac{1}{v}\frac{\partial v}{\partial x}, \\ \frac{dq}{dt} &= -\frac{1}{v}\frac{\partial v}{\partial z}. \end{aligned} \tag{1}$$

We assumed that the solution of the system (1) is defined on a small interval from t to $t+2h$ and x, z are inside a single layer. That means that velocity $v(x,z)$ and its first partial derivatives, $\frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial z}$, are continuous functions for all t inside the interval.

For convenience rewrite four differential equations (1) in the vector notation

$$\frac{d\mathbf{Y}}{dt} = \mathbf{F}(t, \mathbf{Y}) \tag{2}$$

where $\mathbf{Y} = \{y^1, y^2, y^3, y^4\}^T = \{x(t), z(t), p(t), q(t)\}^T$ and $\mathbf{F} = \{f^1, f^2, f^3, f^4\}^T$.

For solving (2) we use the following classical *fourth-order* one-parametric family of the Runge-Kutta formulas

$$\begin{aligned} k_1^i &= hf^i(t, y^i), \\ k_2^i &= hf^i\left(t + \frac{h}{2}, y^i + \frac{k_1^i}{2}\right), \\ k_3^i &= hf^i\left(t + \frac{h}{2}, y^i + \left(\frac{1}{2} - s\right)k_1^i + sk_2^i\right), \\ k_4^i &= hf^i\left(t + h, y^i + (s-1)k_2^i + sk_3^i\right), \\ \Delta y^i &= \frac{1}{6}\left(k_1^i + (4-2s)k_2^i + 2sk_3^i + k_4^i\right) \end{aligned} \tag{3}$$

with the parameter $s = \frac{2+\sqrt{2}}{2}$. Here i=1,2,3,4.

To estimate the accuracy of the integration scheme (3) we should exert some adaptive control over its own progress, making frequent changes in its stepsize. With fourth-order Runge-Kutta, the most straightforward technique by far is the *step doubling* (see, e.g., [1]). We take each step twice, once as a full step, then, independently, as two half steps.

Let us denote the exact solution for an advance from t to $t+2h$ by $y^i(t+2h)$ and the two approximate solutions by y_1^i (one step $2h$) and y_2^i (2 steps each of size h). Since the basic method is of the fourth order, the true solution and the two numerical approximations are related by

$$\begin{aligned} y^i(t+2h) &= y_1^i + (2h)^5\phi^i + O(h^6) + \ldots \\ y^i(t+2h) &= y_2^i + 2(h)^5\phi^i + O(h^6) + \ldots \end{aligned} \tag{4}$$

where, to order h^5 , the value ϕ^i remains constant over the step. (Taylor series expansion tells us the the values of ϕ^i is of the order of ${y^i}^{(5)}(t)/5!$.) The first expression in (4) involves $(2h)^5$ since the step size is $2h$, while the second expression involves $2(h^5)$ since the error on each step is $h^5\phi^i$. The difference between the two numerical estimates is a convenient indicator of truncation error

$$\Delta^i \equiv y_2^i - y_1^i \tag{5}$$

It is this difference that we shall endeavor to keep to a desired degree of accuracy, neither too large nor too small. We do this by adjusting h.

Also, by ignoring terms of the order h^6 and higher, we can solve the two equations in (4) to improve our numerical estimate of the true solution $y^i(t+2h)$, namely,

$$y^i(t+2h) = y_2^i + \frac{\Delta^i}{15} + O(h^6) \tag{6}$$

This estimate is accurate to the fifth order, one order higher than the original Runge-Kutta steps. However, even though (6) may be accurate to the fifth order,we have no way of monitoring its truncation error. Thus higher order is not always higher accuracy! Use of (6) rarely does harm, but we have no way of directly knowing whether it is doing any good. Therefore we should use Δ^i as the conservative error estimate, while keeping in mind potential accuracy gain derived from (6).

Nevertheless, we are using very smooth representation of the velocity function (bicubic spline interpolation) and fifth derivative of y^i is nearly constant. So, for our goal we are using double stepping procedure together with the fifth order truncation accuracy until the average sum of components' error $|\Delta^i|$ divided by 15 is less the then small accuracy constant, ε, namely,

$$\frac{1}{15}\sum_{i=1}^{4}\frac{|\Delta^i|}{4} < \varepsilon \tag{7}$$

12.5. Coordinate Systems

12.5.1. Cartesian coordinate systems. We consider a system of general right-handed Cartesian coordinates (x, y, z). The origin is placed at the Earth's center, z axis is directed to the north, x axis passes throw the intersection of the equator and the zero meridian. The y axis is orthogonal to x and z and forms a right-handed coordinate system.

12.6. Geographic/geocentric spherical coordinate system

We introduce the geocentric spherical coordinates $(x, y, z) = (\varphi, \lambda, r)$ as follows: φ is the latitude (positive northward) $(-\frac{1}{2}\pi \le \varphi \le \frac{1}{2}\pi)$, λ is the longitude (positive eastward) $(0 \le \lambda < 2\pi)$ and $r \ge 0$ is the radial distance.

These geocentric spherical coordinates are related the above Cartesian coordinates in the following way:

$$x = r\cos\varphi\cos\lambda,\ y = r\cos\varphi\sin\lambda,\ z = r\sin\varphi$$

For our purposes, e.g. for model description, we will use depth,H, instead of radial distance, r, which is defined as: $H = R_0 - r$ where R_0 is a constant, usually equal to the Earth's radius.

Geographic spherical coordinates are widely used in geophysics and seismology. Longitude, radial distance have the same sense as for geocentric coordinates. However, φ_{geogr} is an angle between normal to the earth ellipsoid and the equatorial plane at a local point. For geocentric coordinates latitude φ is an angle between a normal to the sphere and the equatorial plane. They are connected via a relation:

$$\varphi = \arctan(\tan(\varphi_{geogr}) \cdot 0.993277)$$

12.6.1. Spherical polar coordinate system. We introduce the spherical polar coordinates $(x, y, z) = (\theta, \lambda, r)$ as follows: θ is the colatitude (positive southward), $0 \leq \theta \leq \pi$, λ is the longitude (positive eastward) $(0 \leq \lambda < 2\pi)$ and $r \geq 0$ is the radial distance. We will also use the depth H instead or the radial distance r, where $H = R_0 - r$. Spherical polar coordinates are related to the Cartesian ones by the following formulas:

$$x = r \sin\theta \cos\lambda, \ y = r \sin\theta \sin\lambda, \ z = r \cos\theta$$

12.6.2. Spherical coordinate system with an arbitrary origin. This coordinate system is used for SSSC's representation. This system is look like spherical polar coordinate system with depth but the origin is placed at any arbitrary point on a sphere with radius R_0. More detail. Let, $(x, y, z) = (\theta, az, H, A)$, where A is an arbitrary point on a sphere. Define coordinates in the following way: θ is a central angle between A and (x, y, z) point, az is azimuth at A to (x, y, z), H is a depth of (x, y, z) point.

References

Gear, C.W. 1971, Numerical Initial Value Problems in Ordinary Differential Equations (Englewood Cliffs, NJ: Prentice-Hall). [1]

Gjøystdal, H., Reinhardsen, J.E. and Åstebøl, K. (1984). A flexible user-oriented system for 3D dynamic ray tracing in complex geological structures. NORSAR Contributions 331.

Cerveny, V., Klimes, L. and Psencik I. (1988). Complete Seismic-ray tracing in three-dimensional structures. In *Seismological Algorithms. Computation methods and computer programs.* Edited by Durl J. Doornbos,Academic Press, p89-168.