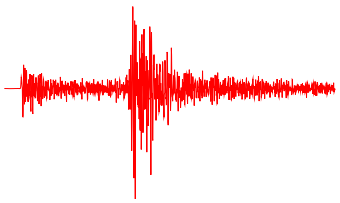
COMPUTER PROGRAMS IN SEISMOLOGY



AN OVERVIEW OF SYNTHETIC SEISMOGRAM COMPUTATION

Robert B. Herrmann
Editor

Professor of Geophysics
Department of Earth and Atmospheric Sciences
Saint Louis University

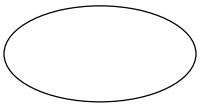
Version 3.30
2002 **(Updated January, 2025)**

Contents

PREFACE

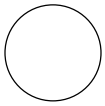
Preparing the latest version of Computer Programs in Seismology included a major effort toward making the programs easier to use. This is done in part by reducing the number of programs in the distribution and eliminating those designed for algorithm verification. The choice of programs to eliminate was based on the usage of programs during the past decade. Another change was an effort to provide consistency in file formats and command line flags.

In this volume I describe programs used for synthetic seismogram generation. Simple flowcharts are given to indicate the logical flow and the program output. The following symbols are used:

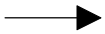


data format

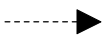
program



other output/input



processing flow

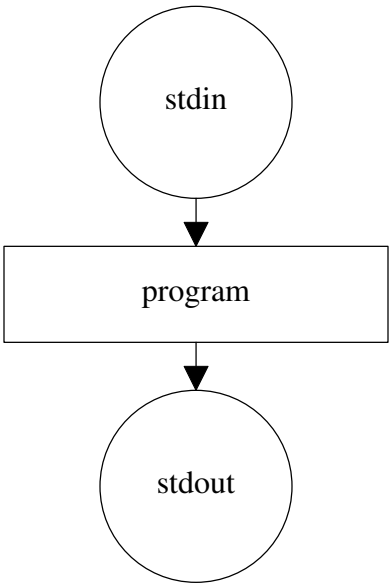


optional processing flow

Since the programs will be run on UNIX and MSDOS systems, command line redirection in place of terminal entry is used to read data into the program from a file. This standard entry in the C language is **stdin**. The corresponding standard program output is **stdout**. This terminology will be used in flow charts. A circle with **stdin** indicates input from a terminal or from a file using the input redirection, and a circle with **stdout** indicates output to the terminal, or output to a user file using output redirection. A simple instance of redirection is

```
rbh> program < input_file > output_file
```

Here, as in all examples, the **rbh>** indicates a command line prompt. This program usage will be represented in a flow diagram as



Even though almost all programs are written in FORTRAN, ASCII files that are used for standard data formats are defined to be parsed. Thus, if several numbers are written per line, they will be separated by white space. In addition, if an array is written, e.g., a time series with five values per ASCII line, then the last entry will be zero filled so that a consistent C format can be used, e.g., " %f %f %f %f %f ".

A reference to a specific documented program or file format is indicated in bold, e.g., **genray96**. Specific input/output files are indicated in italic, e.g., *RAY96.PLT*. Terminal input/output is indicated by a courier font, e.g.,

`genray96 -d dfile`.

I begin with a discussion of data formats, followed by a discussion of individual programs for generating synthetic seismograms.

Finally, I describe the program usage, but the leave detailed documentation in the manual pages for the individual volumes. In addition, the ultimate source of what a program can do is the actual source code.

Update June, 2021

As new capabilities were added to the codes, the documentation did not describe all new command line options. In addition the demise of Tektronix grpahics terminals and simulations meant that support to such graphics could be removed. X11 graphics are still useful and graphics format converter packages such as Imagemagick make it possible to create PNG files for the web from Encapsulated PostScript files. Finally introduction of **`gsac`** means that most of the *file96* filter codes are rarely used.

CHAPTER 1
DATA FORMATS

1. Introduction

The focus of the previous version of *Computer Programs in Seismology* was the implementation of specific algorithms for the analysis of seismic data. Little effort was directed toward developing a consistent set of input and output files for these programs. The focus was on specific volumes for each algorithm.

The use of these programs for the analysis of seismic data identified that more focus should be placed on data/model formats rather than on the algorithms. The two basic files used for all synthetic seismogram computations are the earth model file and the output time history. Thus the user will find the programs easier to use if different algorithms can be run using the same input and output formats. The specification of these file formats is complicated by the many different ways the earth models can be described and seismic data can be stored.

All programs described here will use a **model96** earth model description and will create a time history in the **file96** format. The earth model format is well defined for layered isotropic media, but is general enough to encompass other more complicated models. The output file format is general enough for the present uses of data, can be converted easily to other formats, and can be extended in the future. The basis of this flexibility is the use of keyed comment lines at the head of each file and the allowance for future extension.

An example of how this concept works is given in Figures 1 and 2. Figure 1 indicates that the same earth model file can be used to generate synthetic seismograms by generalized ray, model summation or wavenumber integration. Figure 2 indicates that certain filtering operations have been designed to operate on **file96(V)** files.

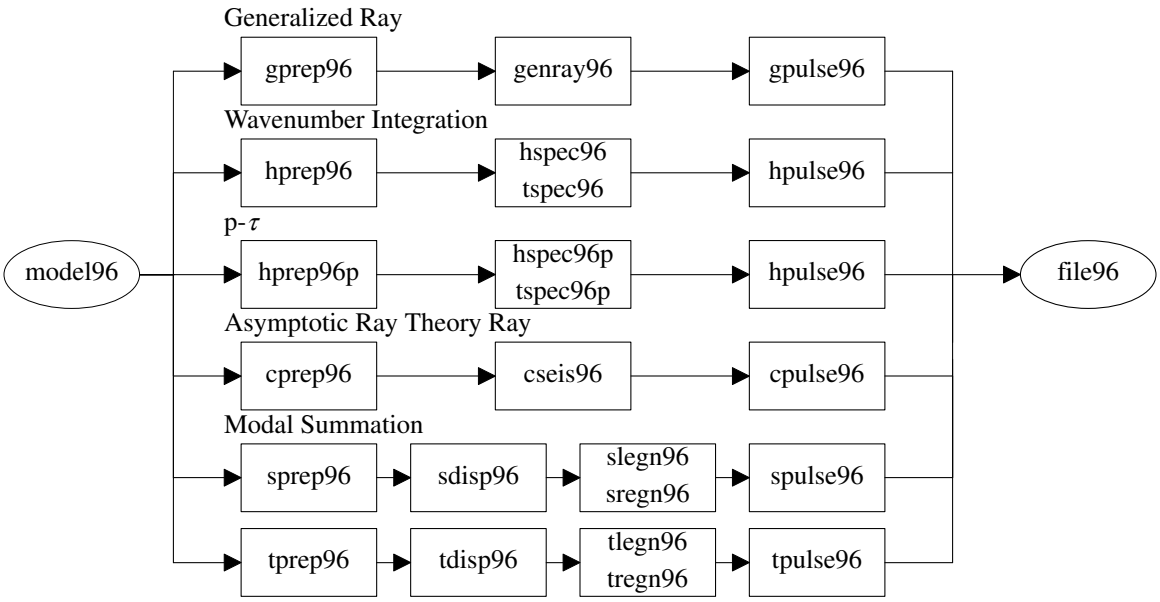


Fig. 1. Processing flow for synthetic seismogram programs.

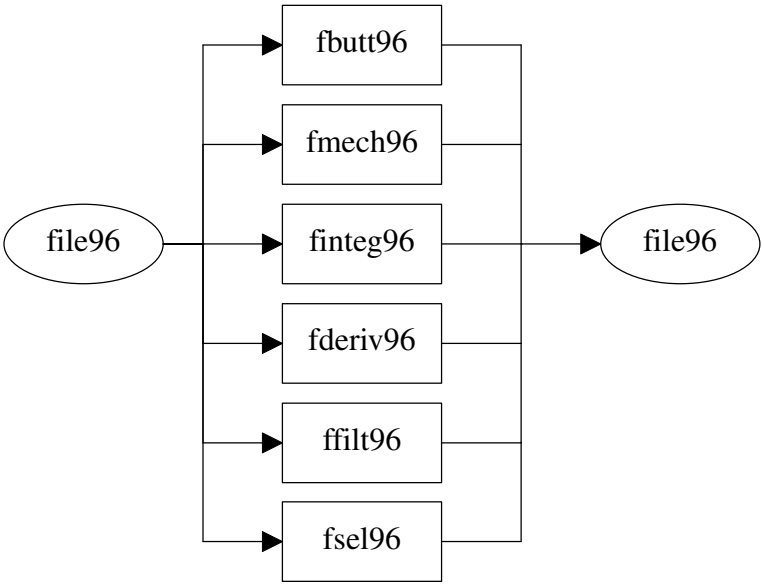


Fig. 2. Filtering operations.

2. Earth Model

A **model96** model file consists of eleven ASCII header lines followed by the detailed model. An example is the best way to present this concept. The model shown below is for a flat, isotropic, flat-layered, constant velocity earth.

```
MODEL.01
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
 H       VP      VS  RHO QP  QS  ETAP ETAS FREFP FREFS
40.     6.0     3.5 2.8 0.0 0.0 0.0  0.0   1.0   1.0
00.     8.0     4.7 3.3 0.0 0.0 0.0  0.0   1.0   1.0
```

The context of the lines in this example are as follow:

Line 01: The *first* line requires the use of the keyword **MODEL** which identifies this ASCII file as an earth model file. This keyword is absolutely necessary. In this example the keyword is **MODEL.01** indicates that the format is Version 01.

Line 02: The *second* line is a descriptive comment for the model. Its only limitation is that it can be no more than 80 characters long.

Line 03: The *third* line describes the isotropy of the medium. The allowed keywords are **ISOTROPIC**, **TRANSVERSE ISOTROPIC** and **ANISOTROPIC**. All programs in *Computer Programs in Seismology* will work with the **ISOTROPIC** designation.

Line 04: The *fourth* line indicates the model units. **KGS** indicates that the distances, layer thicknesses are in *kilometers*, velocities are in *kilometers/second*, the density is in gm/cm^3, and that time is in *seconds*. The units are necessary for defining synthetic seismogram amplitudes.

Line 05: The *fifth* line uses the key words **FLAT EARTH** or **SPHERICAL EARTH**.

Line 06: The *sixth* line describes the nature of model boundaries, e.g., **1-D**, **2-D**, or **3-D**,

Line 07: The *seventh* line describes the nature of layer velocities, e.g., **CONSTANT VELOCITY** or **VARIABLE VELOCITY**.

Lines 08-11: The *eighth* through *eleventh* lines are positions left for future extension in the file format.

Remaining Lines: The remaining lines are specific to a particular model format.

For an isotropic, one-dimensional medium with constant velocity, the earth model consists of a sequence of flat layers in the **FLAT EARTH** or spherical shells in the **SPHERICAL EARTH**. The first line of this model definition is an ASCII string

indicating column headings. This is permitted for readability, but the order of the actual numerical entires is fixed. The remaining lines specifically define the model. The model is described in free format by 10 columns:

$$\mathbf{H} \quad \mathbf{V_P} \quad \mathbf{V_S} \quad \rho \quad \mathbf{Q_P} \quad \mathbf{Q_S} \quad \eta_\mathbf{P} \quad \eta_\mathbf{S} \quad \mathbf{f_P} \quad \mathbf{f_S}$$

where H is the layer thickness, with the bottom layer thickness being 0 or more kilometers. If H is negative at the top of the layer stack, then this will define a reference level of the surface (this will facilitate studying earth - atmosphere interaction). V_P and V_S are the compressional and shear wave velocities in each layer, and ρ is the layer density. The anelastic attenuation model is described in terms of three parameters: Q_v, η_v and f_v, where v is either P or S. For wavetype v, the specific quality factor as a function of frequency f is given by $Q_v(f) = Q_v(f / f_v)^{\eta_v}$.

In the computer programs, $Q < 1$ is never permitted, which is not an unreasonable assumption. Thus the earth model may be given in terms of Q or Q^{-1}, since the programs will automatically associate a value < 1 as being Q^{-1} and a value > 1 as being Q. This will make data input easier.

Finally, the documentation of a specific program will indicate the suite of models that can be considered.

To assist in creating the file correctly, Chapter 8 discusses the program **`mkmod96`**.

3. Time histories

Time histories are stored in the **file96** format. Earlier versions of the package had two time history formats: one for Green's functions of synthetic seismograms, and other for three-component time histories. Because of this, it was necessary to have duplicates of algorithms, e.g., integration, filtering or differentiation, to act on the two time history formats. This led to a set of programs differing only in input/output formats.

Other seismological formats exist. The SAC format file consists of a single trace, which can be inconvenient and inefficient for storing Green's functions, or typical three-component seismic time histories. Both SAC headers and the CSS data base provide information about many aspects of the time history, whereas the earlier versions of this package did not.

The approach taken here is to implement general purpose programs that work on time histories, irrespective of the specific time history format. Thus one will be able to filter or display Green's functions or specify a source mechanism to convert the Green's functions to a three-component time history and then filter or display the traces, using the same programs.

The cost for generality is a slightly more complicated file format, **file96**.

Consider the following example:

The format consists of two sections: a station header and specific traces, each of which has an individual trace header as well as the trace itself. The meaning of the lines in this example are as follow:

```
FILE21.02
SYNTHETIC
TIME_DOMAIN
COUNTS
None
    0  0  0  0  0     0.0000   0.000000    0.000000  10.000000
GRN16
   0.000000    0.000000      0.000
        10.0000          0.0899          0.0000        180.0000
None
model.d
Pa
     2.357022524         4.040610313         4.040610313
  0.1008E+03  0.1008E+03  0.3220E+02  0.3430E+02  0.3430E+02  0.2800E+01
LINE15
LINE16
  1  4  1  4  5  1  4  5  1  4  1  4  1  4  5  0  0  0  0  0  0
ZEX
    -90.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
REX
      0.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
ZVF
    -90.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
RVF
      0.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
ZHF
    -90.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
RHF
      0.000      0.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
THF
      0.000     90.000      0.125     2
    0  0  0  0  0     0.6667
  0.00000000E+00   0.00000000E+00   0.00000000E+00   0.00000000E+00
```

Station Header

Line 01: The *first* line requires the use of one of the following keywords: **FILE01.02**, **FILE03.02**, or **FILE16.02**, which uniquely defines the file as being a general time series format. The keywords indicate that the ASCII file is a single component, three component or sixteen component time history,

respectively. In the case of multiple components, not all need be present, as in the case of only transverse component Green's functions. The **.01** extension indicates the version number of the current format definition, with the current version being 01. One of these keywords is required.

Line 02: The *second* line requires the use of one of the two keywords: **OBSERVED** or **SYNTHETIC**. This is used to identify the origin of the trace, and also to resolve issues concerning the time of the first sample. One of these keywords is required.

Line 03: The *third* line requires the use of one of the two keywords: **TIME_DOMAIN** or **FREQUENCY_DOMAIN** to indicate whether the series is a time series or a complex Fourier spectra. At present, only the **TIME_DOMAIN** is implemented. One of these keywords is required.

Line 04: The *fourth* line indicates the units of the time series. For **TIME_DOMAIN** the current options are **COUNTS**, **CM**, **CM/SEC**, **CM/SEC/SEC**, **M**, **M/SEC**, **M/SEC/SEC**, **MICRON**, **MICRON/SEC**, or **MICRON/SEC/SEC**, One of these keywords is required.

Line 05: The *fifth* line is a comment string of no more than to 80 characters in length describing the filtering operations performed on the data stream. If no processing has been done, the keyword is **NONE**.

Line 06: The *sixth* line information on the event origin, if known. The entries in space separated free format are
YEAR (an integer, e.g., 1990),
MONTH (an integer, e.g., 12),
DAY (an integer, e.g., 28),
HOUR (an integer, e.g., 13),
MINUTE (an integer, e.g., 6),
SECOND (a float, e.g., 23.345),
EVENT_LATITUDE (a float, north = positive),
EVENT_LONGITUDE (a float, east = positive), and
EVENT_DEPTH (a float, positive = down).
For **SYNTHETIC**, all entries are zero, except **EVENT_DEPTH**.

Line 07: The *seventh* line provides information on the station name. The line consists of a character string for the station name:
STATION_NAME (character*8),
For **SYNTHETIC**, **STATION_NAME** is **GRN16**,

Line 08: The *eighth* line provides information the station location. Space separated free format is used. The order of entries on the line are
STATION_LATITUDE (a float, north = positive),
STATION_LONGITUDE (a float, east = positive),
STATION_ELEVATION (a float in kilometers, positive = up).

For **SYNTHETIC**, **STATION_LATITUDE** and **STATION_LONGITUDE** are 0.0, and **STATION_ELEVATION** is receiver depth in kilometers (down is positive).

Line 09: The *ninth* line provides information about the position of the station with respect to the source, if known. The entries in space separated free format are
DISTANCE_KILOMETERS (distance between source and receiver,
DISTANCE_DEGREES (distance in degrees, roughly 111.195°/km)
STATION_EVENT_AZIMUTH (azimuth in °, North=0, East= 90)
EVENT_STATION_AZIMUTH (back azimuth in °, North=0, East= 90).
All four fields are required.

Line 10: The *tenth* line is a comment string up to 80 characters in length describing the source pulse used in making the synthetic seismograms. If no convolution with a source time function has been performed, the keyword **NONE** is used.

Line 11: The *eleventh* line is a comment line. For **SYNTHETIC** time histories this will be the name of the earth model file.

Line 12: The *twelvth* line indicates the units of the pressure or stress time series. The current options are **Pa**, **MPa**, One of these keywords is required.

Line 13: The *thirteenth* line gives the model predicted ray theory first arrival times of P, SV and SH for the model. If these are not defined, each is set to *-12345.0*, which is SAC syntax for an undefined number. The purpose of this field is to be able to define the first arrival times in the SAC file for use in source parameter inversion.

Line 14: The *fourteenth* line gives the model predicted medium parameters A, C, F, L, N and ρ at the source depth. These are the five medium parameters for a transversely isotropic medium with vertical axis of symmetry and the density. For an isotropic medium, $A = C = \lambda + 2\mu$, $F = \lambda$, and $L = N = \mu$. The units chosed here assume that the corresponding wave velocty is in *km/sec* and the density is gm/cm^3. If the the data file corresponds to observed and not synthetic time series, or if these are not defined, then all six parameters are set to 0.0!

Lines 15-16: The *fifteenth* through *sixteenth* lines are positions left for future expansion of the file format.

Line 17: The *seventeenth* line consists the of the **JSRC** array of 21 integers. The purpose of this array is twofold.

First, if the *i*'th trace does not exist, or is not generated, then **JSRC(i) = 0** . This makes this ASCII file smaller in size since only non-zero traces are stored.

The second purpose is to identify the trace type, e.g., 1 = Z (vertical), 2 = N

(north), 3 = E (east), 4 = R (radial), 5 = T (transverse), and 6 = O (other). This second purpose is only introduced to make later manipulation of traces easier. The ultimate definition of trace orientation is in the individual trace header.

Trace Header and Trace

There are a maximum of 1, 3 or 16 traces associated with the **FILE01**, **FILE03** or **FILE16** keywords, respectively. The presence of a trace is indicated by the **JSRC(i) ≠ 0** flag in the Station Header. Each trace description consists of a three line header followed by the traces.

Line 01: The *first* line contains a character string for the component name:
COMPONENT_NAME (character*8),

Line 02: The *second* line gives component orientation and sampling information:
COMPONENT_INCIDENCE (float that describes the angle of positive motion with respect to the vertical. A value of -90 indicates up, 0 indicates horizontal, and 90 indicates down)
COMPONENT_AZIMUTH (float that describes horizontal orientation of positive motion, e.g., 0 is north, 90 is east)
COMPONENT_SAMPLE_INTERVAL (float describing sampling interval in seconds),
COMPONENT_NUMBER_OF_SAMPLES (integer)

Line 03: The *third* line uses space separated free format to define the time of the first sample:
YEAR (an integer, e.g., 1990),
MONTH (an integer, e.g., 12),
DAY (an integer, e.g., 28),
HOUR (an integer, e.g., 13),
MINUTE (an integer, e.g., 6),
SECOND (a float, e.g., 23.345),

Remaining Lines: The remaining lines consist of the trace written in the FORTRAN format *(e16.8,1x,e16.8,1x,e16.8,1x,e16.8)*. There are four entries per line. If number of points is not a multiple of 4, then the line is zero filled. *(Beware of differences in the E notation in FORTRAN and C for different compilers. In FORTRAN on usually will see and entry like +1.000E+21 and in C an entry like +1.000E+021. This make cause difficulty if the user attempts to duplicate this format in C.)*

4. Distance File

This file defines characteristics of the time series to be created. It consists of five ASCII entries per line in free format. These entries are

```
DIST DT NPTS T0 VRED
```

where **DIST** is the desired epicentral (horizontal distance) from the source to the receiver, **DT** is the sampling interval in seconds, **NPTS** is the number of samples which must be a power of 2 (note the programs will automatically check this), and the time of the first sample is
T0 + DIST/VRED seconds. If **VRED = 0**, then the time of the first sample is **T0**. For use by 2-D ray tracing programs, **DIST** can be negative.

This issue of distance may be revisited later when 2-D models are fully incorporated, since one may wish to specify the horizontal position of the source as well as the receiver.

Some programs, e.g., **hprep96(VI)** and **hprep96p(VI),** will read the distance file in its entirety and attempt to determine a common **DT**, usually the smallest, and **NPTS**, usually the largest. The time series output values of **DT** and **NPTS** may thus differ from that requested. This is because of the need to sample exactly the same frequencies for all distances. If this result is not desired, the user should execute the programs separately for each distance.

A sample distance file is

```
 10.0     0.125   256    -1.0 6.0
 20.0     0.125   256    -1.0 6.0
 30.0     0.125   256    -1.0 6.0
 40.0     0.125   256    -1.0 6.0
 50.0     0.125   256    -1.0 6.0
 60.0     0.125   256    -1.0 6.0
 70.0     0.125   256    -1.0 6.0
 80.0     0.125   256    -1.0 6.0
 90.0     0.125   256    -1.0 6.0
100.0     0.125   256    -1.0 6.0
```

5. Depth File

This file defines the source or receiver depths in the case that synthetics for more than a single depth are to be generated. In the programs **hprep96(VI)** for wavenumber integration, and **hprep96p(VI)** for p-τ response, multiple depths for source or receiver is indicated by the command line flags **-FHS source_depth_file** or **-FHR receiver_depth_file**, respectively.

The contents of the depth file consist of a single ASCII entry per line in free format giving the depth:

```
depth
```

A sample depth file is

```
 0.0
 2.5
 5.0
 7.5
10.0
12.5
15.0
17.5
20.0
```

6. SURF96

The **surf96** format is used for experimental dispersion data and is generated by the program **sdpegn96** and used by **sdpegn96** and **sdpdsp96**. The format is very simple:

```
SURF96 WAVE TYPE FLAG MODE PERIOD VALUE DVALUE
 ....  .... .... .... .... ....  ...   ....
SURF96 WAVE TYPE FLAG MODE PERIOD VALUE DVALUE
```

where

SURF96	a left-justified keyword,
WAVE	single character **R** i **L** for Rayleigh or Love
TYPE	single character **C**, **U**, or **G** for the observed phase velocity, group velocity or gamma
FLAG	single character **X** for observed or **T** for theoretical value. This distinction is used in plotted theoretical dispersion curves and observed data
MODE	mode of observation: **0** is fundamental mode, **1** is 1st
PERIOD	period of observation in units of *seconds*
VALUE	observed dispersion in units for *km/sec* for **C** or **U** or km^{-1} for **G** type
DVALUE	error in observed value. Note for synthetics, this is set to a small value of *0.001* for velocity and *0.0000001* for gamma. For observed data, these values should be be standard error of the mean and not the standard error of the observation.

A sample **SURF96** dispersion file is

```
SURF96 R C X   0    5.0000        3.2044       0.10000E-02
SURF96 R U T   0    5.0000        3.0707       0.10000E-02
SURF96 R G T   0    5.0000       0.31785E-03   0.10000E-06
SURF96 L C T   0    8.0000        3.6255       0.10000E-02
SURF96 L U T   0    8.0000        3.4402       0.10000E-02
SURF96 L G T   0    8.0000       0.16573E-03   0.10000E-06
```

As with all formats, except the **Depth File** format, the keywords are a required part of the

file definition. In addition the **L**, **R**, **C**, **U** and **G** codes are used for benefit of people and not computers.

7. SAC

SAC is a standard trace analysis package for manipulating time series. *Computer Programs in Seismology*.

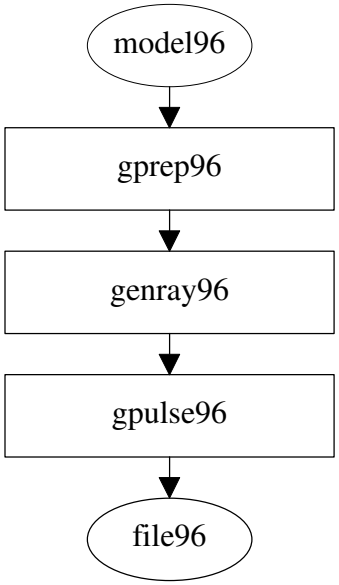
Several conversion programs, described in detail in Chapter 6, were written to work with SAC files:

sactoasc - convert SAC binary to SAC ascii format
asctosac - convert SAC ascii to SAC binary format
This is useful in transferring SAC files between machines of different binary architecture.
shwsac - reads a SAC binary file, shows all header variables, and lists ten representative values of the times series.
f96tosac - convert **file96(V)** to SAC binary format
sactof96 - convert SAC binary to **file96(V)** format
saclhdr - list the SAC header value for use in SHELL scripts.
sacdecon - a water level deconvolution program using SAC files.
saciterd - iterative time domain deconvolution program using SAC files.
sacevalr - instrument response deconvolution using amplitude and phase response files generated by the IRIS program *evalresp*.
sacfilt - apply or remove a filter response defined by a SAC pole-zero file
saccvt - convert SAC file from big-endian (SPARC) to little-endian (INTEL) and vice versa.

CHAPTER 2
GENERALIZED RAY REVISED

1. Introduction

In this chapter I describe the use of generalized rays for the generation of synthetic seismograms. This type of seismogram synthesis is very efficient if only a few rays between the source and receiver are required in simple layered media. The difficulties in using more rays for a detailed structure are twofold. First, the ray description is tedious, since the program requires each ray segment to be defined in terms of ray type. The second problem is that computations with many rays may take much longer than using modal summation or wavenumber integration techniques.



To address the first problem, the program **gprep96** is used to automatically generate the ray description file. This output is then used by the program **genray96** to compute the medium response. The output of **genray96** is in the **file96** Green's function format, but convolution with the second derivative of the source time function is done by the program **gpulse96**.

2. gprep96

The generalized ray technique requires the complete specification of a ray path between the source and receiver. This means that one must know whether the ray leaves the source in an upward or downward section, the layer in which each segment of the ray lies, and the wavetype itself for each segment, e.g., P, SV or SH. Such descriptions are not difficult for a simple model with few conversions upon reflection and transmission between P and SV, but become laborious for more complicated models. In addition, changing the layer in which either the source or receiver lie requires a completely different ray specification.

The program **gprep96** automatically provides the ray specifications for all rays between the source and receiver in a manner that makes it trivial to change source or

receiver depths. The secret of the ray generation is to change the point of view from that of rays in layers to one of rays interacting at layer boundaries, with the source and receiver occupying pseudo-boundaries. With this convention, a ray may leave the source going upward (0) or downward (1). The ray may next interact with the adjacent boundary, going either up (0) or down (1). This quickly suggests the use of a binary number system to represent a ray path. If the ray has two segments or legs after leaving the source, then there are four possible rays. Figure 1 illustrates this.

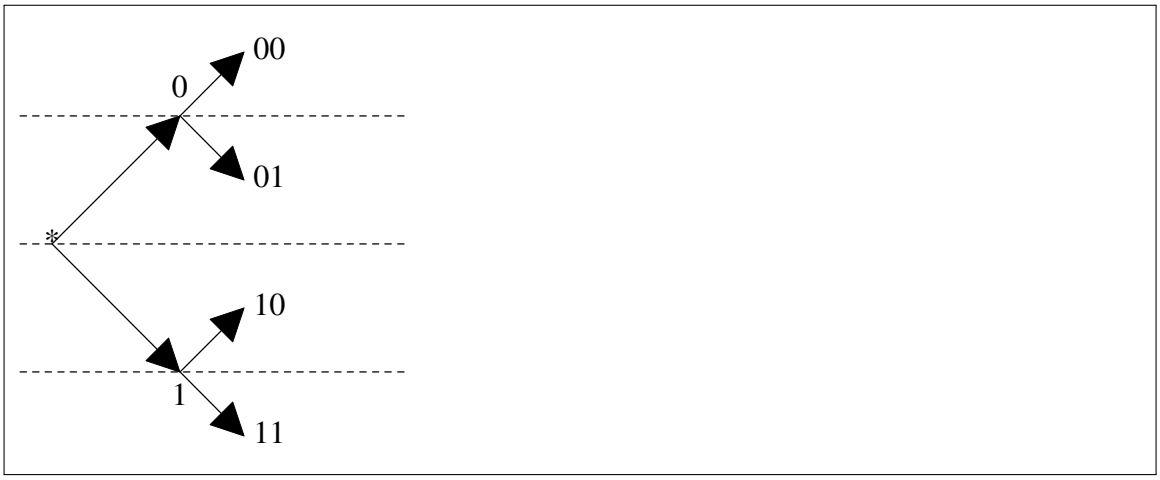


Fig. 1. Simple ray descriptions.

In this picture 4 rays can be represented by two segments. Since a 4 byte integer is used to represent the unique rays in **gprep96**, there can be there can be up to $\mathbf{2^{32}}$ possible rays, e.g., 2 with one segment, 4 with two segments, ..., $\mathbf{2^{31}}$ with 31 segments. The minimum number of segments is the direct ray path between the source and the receiver. If a ray path extends above the top or below the bottom of the model, the particular path is ignored.

Once a ray path is determined, each ray segment can systematically take on 2 values, e.g., P or SV, leading to yet another set of combinations. The maximum number of P-SV segments, permitting conversion upon reflection and refractions, is 4^{32}, which is significantly more than one would ever desire to run.

The large number of possible total rays, as well as the use of a 32 bit integer, means that one may never wish to consider a model with many layers. For example, it would be impossible to use this program with a fifty layer model, with the source and receiver at opposite ends of the model, since 50 segments would be required just for the direct ray and only 31 segments are permitted using the 4 byte representation of integers.

Figure 2 shows the processing flow for this program. The program requires one earth model file in the **model96** format and two optional control files. The output consists of the file *GPREP96.PLT*, a **CALPLOT(I)** graphics file, the file *genray96.ray*, the ray control file for **genray96(V)**, and screen output designated by *stdout*.

Program control is through the command line:

`gprep96` *[flags]*, where the command flags are

-M model - *model* is a file in the **model96** format. This is required.

-DOP

-DOSV

-DOSH

-DOALL

One or more of these must be specified. These tell the program to compute P, SV, SH and all ray segments. If for example, no SH is desired but both P and SV are desired, then use **-DOP -DOSV**. If only **-DOP** is specified, then there will be no P -> SV or SV -> P conversions.

-DOREFL

Permit P-SV conversions only upon reflection, otherwise reflections will only be P $\rightarrow$ P SV $\rightarrow$ SV.

-DOTRAN

Permit P-SV conversions only upon transmission, otherwise transmissions will only be P $\rightarrow$ P SV $\rightarrow$ SV.

-DOCONV

Permit P-SV conversions on both reflection and transmission

The default is that no conversions are computed.

-DENY deny_file

A simple listing of interfaces at which only transmission without wavetype conversion is permitted. Reflections are denied at this boundary. The file consists of a single entry per line, giving the specific boundary number of the earth model. This may be useful to approximate a gradient when there are no turning rays.

-R reverberation_file

A simple listing of layers and the maximum number of ray segments in a layer. This is useful to focus on the reverberations within specific layers. The default is to permit as many as possible.

-N maximum_number_of_segments

The maximum number of ray segments permitted in the ray description. Since artificial boundaries are inserted at the source and receiver depths, this number must be large enough to permit one direct ray between the source and receiver.

-HS source_depth

The depth of the source in the model

-HR receiver_depth

The depth of the receiver in the model

-?

-h

Online help

3. genray96

genray96 creates the synthetic seismogram by applying the Cagniard-de Hoop technique to each ray description. This technique is well described in seismological

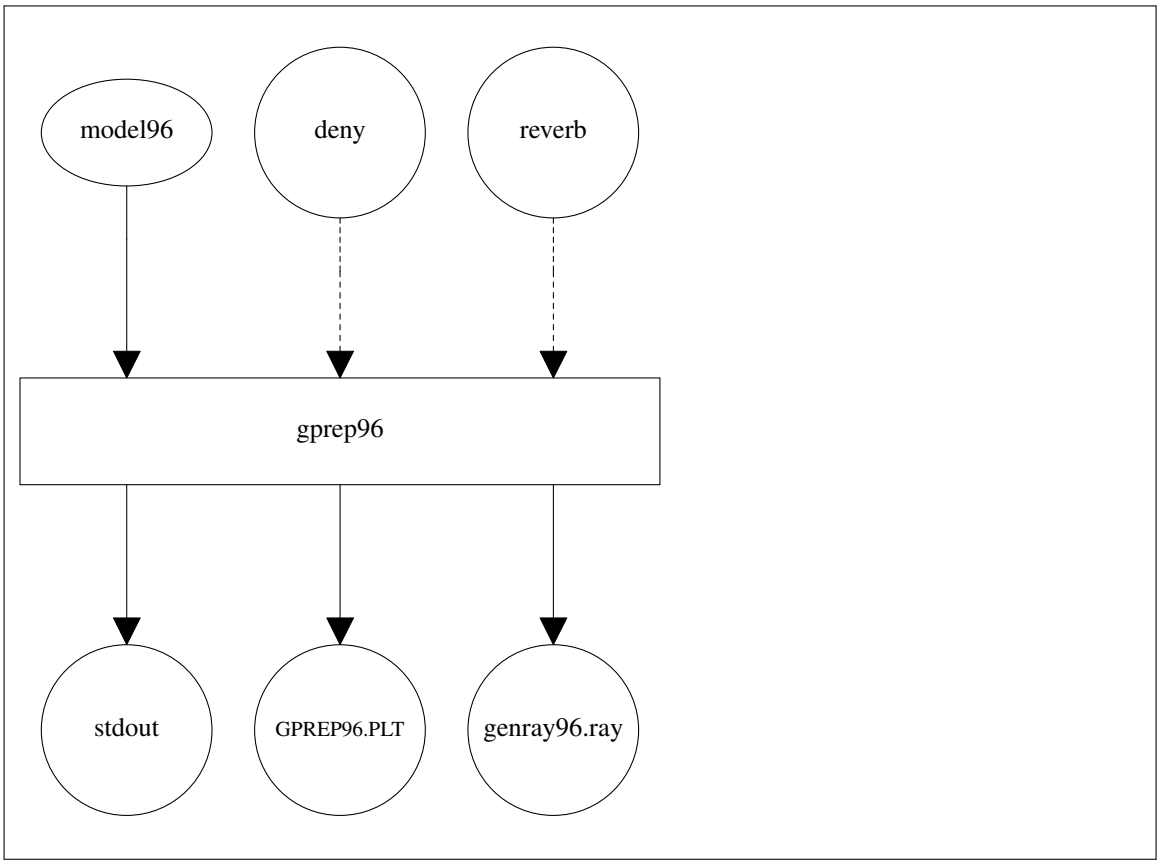


Fig. 2. Processing flow for **gprep96**

literature. This program is a modification of a program originally developed by Dr. C. A. Langston, The Pennsylvania State University, in the late 1970's.

Figure 3 shows the processing flow for this program. The program requires one earth model file in the **model96** format which is referenced through the file *genray96.ray* generated by **gprep96(V)**. The other required input file is a *distance_file* described in Chapter 1. The output consists of the file *genray96.grn* in the **file96(V)** format, a tabulation of travel time for each ray description in the file *genray96.tim* and screen output designated by *stdout*.

Program control is through the command line:

`genray96` *[flags]*, where the command flags are

`-ALL`
Compute all Green's functions *(default true)*.

`-EQEX`
Compute only Green's functions for moment tensor sources. *(default false)*

`-EXF`
Computer Green's functions for explosion and point forces. This addresses exploration sources. *(default false)*

`-d` *dfile*
The required distance file. The file contains the following ASCII entries per line:

DIST DT NPTS T0 VRED

where **DIST** is the epicentral distance in kilometers, **DT** is the sampling interval for the time series, **NPTS** is the number of points in the time series (a power of 2). **T0** and **VRED** are used to define the time of the first sample point which is **T0 + DIST/VRED** if **VRED** $\neq$ **0** or **T0** if **VRED = 0**.

This file is required

-n *nasym*
Number of asymptotic terms to use from the expansion of the modified Bessel function $K_m(spr)$. If *nasym = 1*, near-field contributions are not included. The effect of this term can be seen in Helmberger and Harkrider (1978). *(default nasym= 1)*

-SU
Only produce Green's functions for rays that leave the source upward.

-SD
Only produce Green's functions for rays that leave the source downward.

-SPUP
Only produce Green's functions for P rays that leave the source upward.

-SSUP
Only produce Green's functions for S rays that leave the source upward.

-SPDN
Only produce Green's functions for P rays that leave the source downward.

-SSDN
Only produce Green's functions for S rays that leave the source downward.
(If none of these flags are given, the synthetic will consist of all rays from the source)

-TIME
Do not produce the Green's functions. Just create the file *genray96.tim* which contains travel time information.

-va -vb -vc -vd -ve -vf -vg -vh
Various flags to output intermediate results. Used as a debugging tool. Only attempt to use this with theory and source code at hand.

-?

-h
Program does nothing, other than to list the command line flags.

4. gpulse96

Figure 4 shows the processing flow for this program. The program requires the *genray96.grn* file created by **genray96(V)** and optionally the source pulse definition file *rfile*. The program output is on *stdout* and is a time series in **file96(V)** format. See Appendix B concerning the Green's functions.

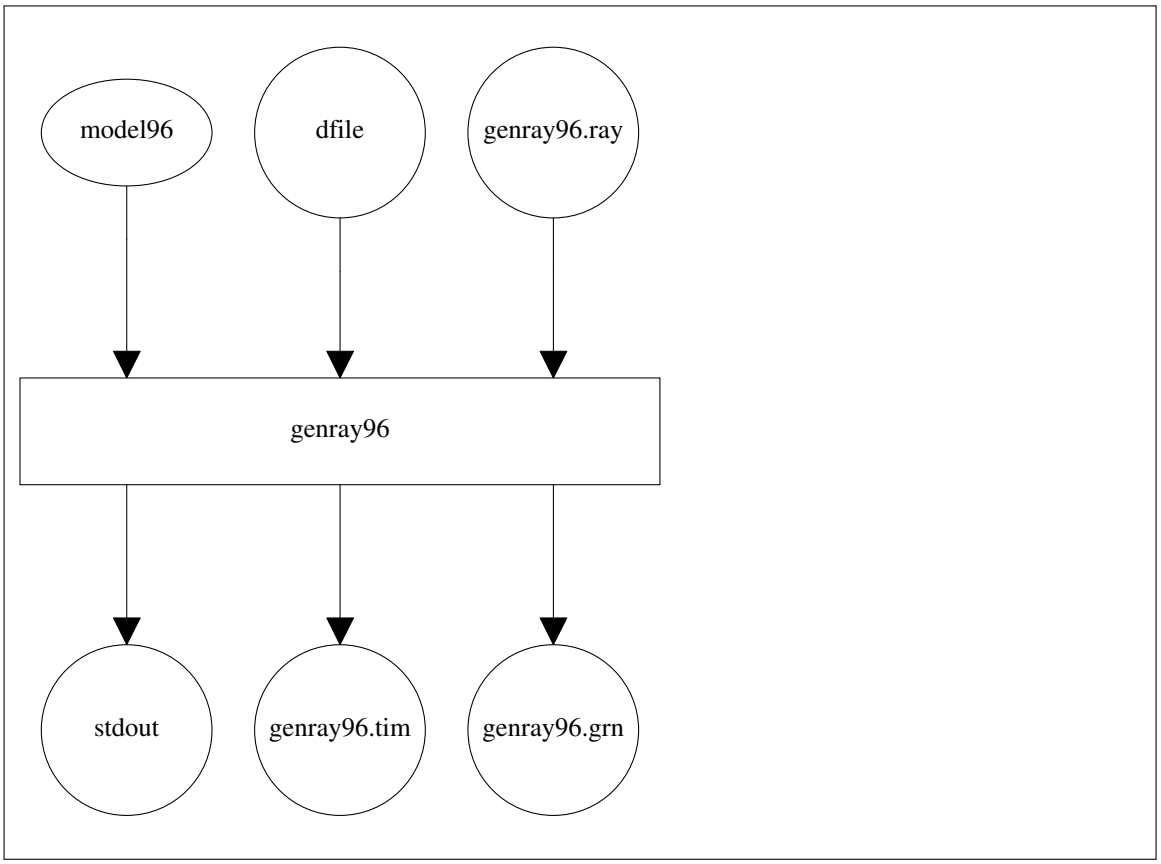


Fig. 3. Processing flow for **genray96**

Program control is through the command line:

`gpulse96` *[flags]*, where the command flags are

`-v`
Verbose output

`-t`
Triangular pulse of base 2 L Δt, which Δt is the sample interval. To avoid problems with sharp truncation in the frequency domain spectra by sampling, never set L < 2. The special case of L = 2 for the triangular pulse is equivalent to the parabolic pulse with L = 1.

`-p`
Parabolic Pulse of base 4 L Δt

`-o`
Ohnaka pulse with parameter alpha

`-i`
Dirac Delta function

`-l L`
Source duration factor for the parabolic and triangular pulses.

`-a alpha`
Shape parameter for Ohnaka pulse

`-D`

Output is ground displacement

-V

Output is ground velocity (default)

-A

Output is ground acceleration

-F rfile

User supplied pulse

-m mult

Multiplier (default 1.0)

-OD

Output is forced to be named displacement

-OV

Output is forced to be named velocity

-OA

Output is forced to be named acceleration

-Z zero phase triangular/parabolic pulse, else causal

-?

-h

Online help concerning program usage

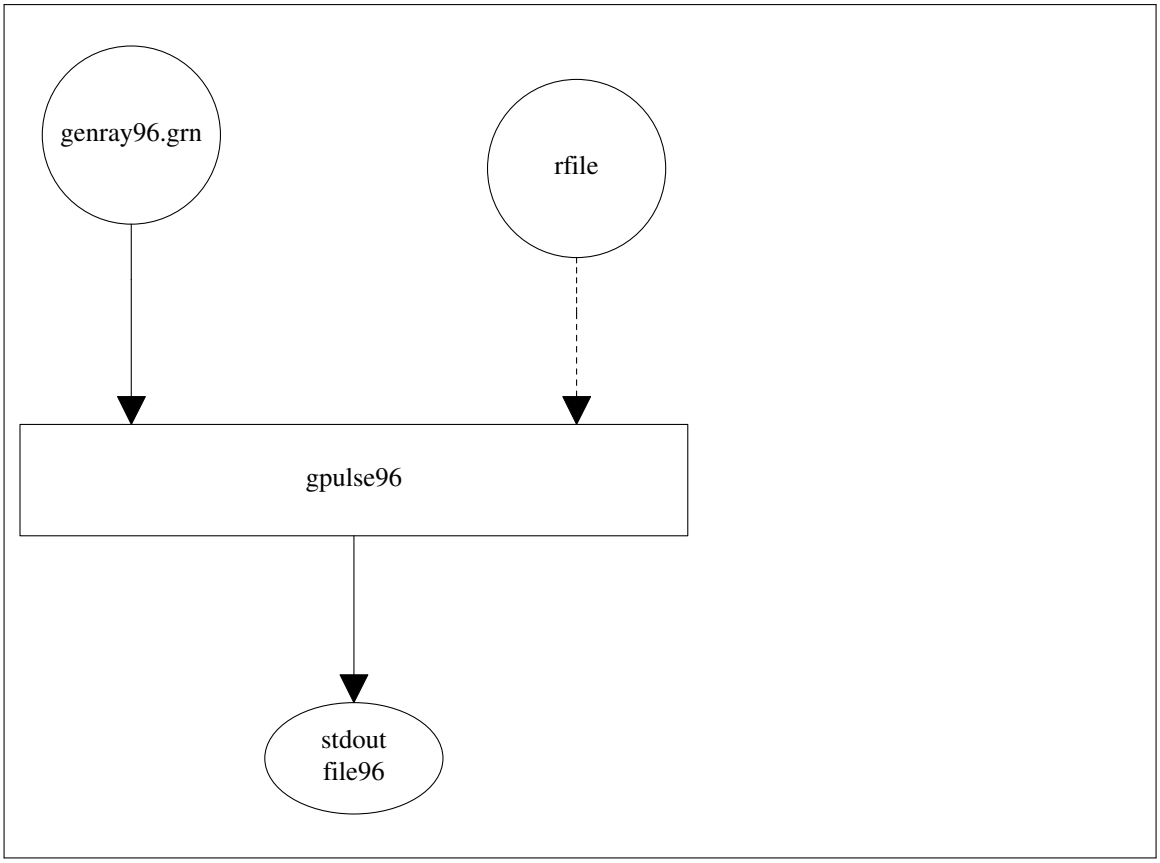


Fig. 4. Processing flow for **gpulse96**

5. Sample Run

Given the sample model and distance files of Chapter 1, the following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV

rbh> genray96 -d dfile   > genray96.out

rbh> gpulse96 -V -p -l 4  | fprof96
```

The graphics output of **gprep96** is given in Figure 5.

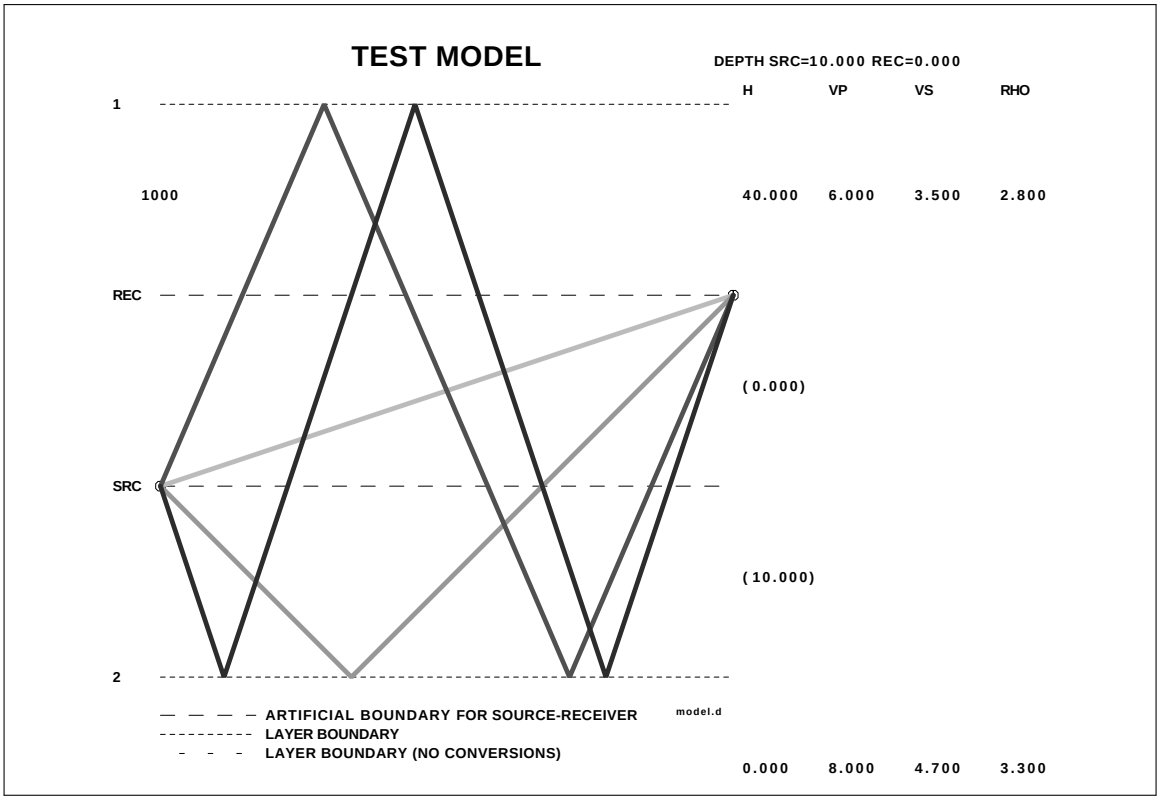


Fig. 5. The GPREP96.PLT generated.

The synthetic transverse time histories for a vertical strike-slip source are given in Figure 6.

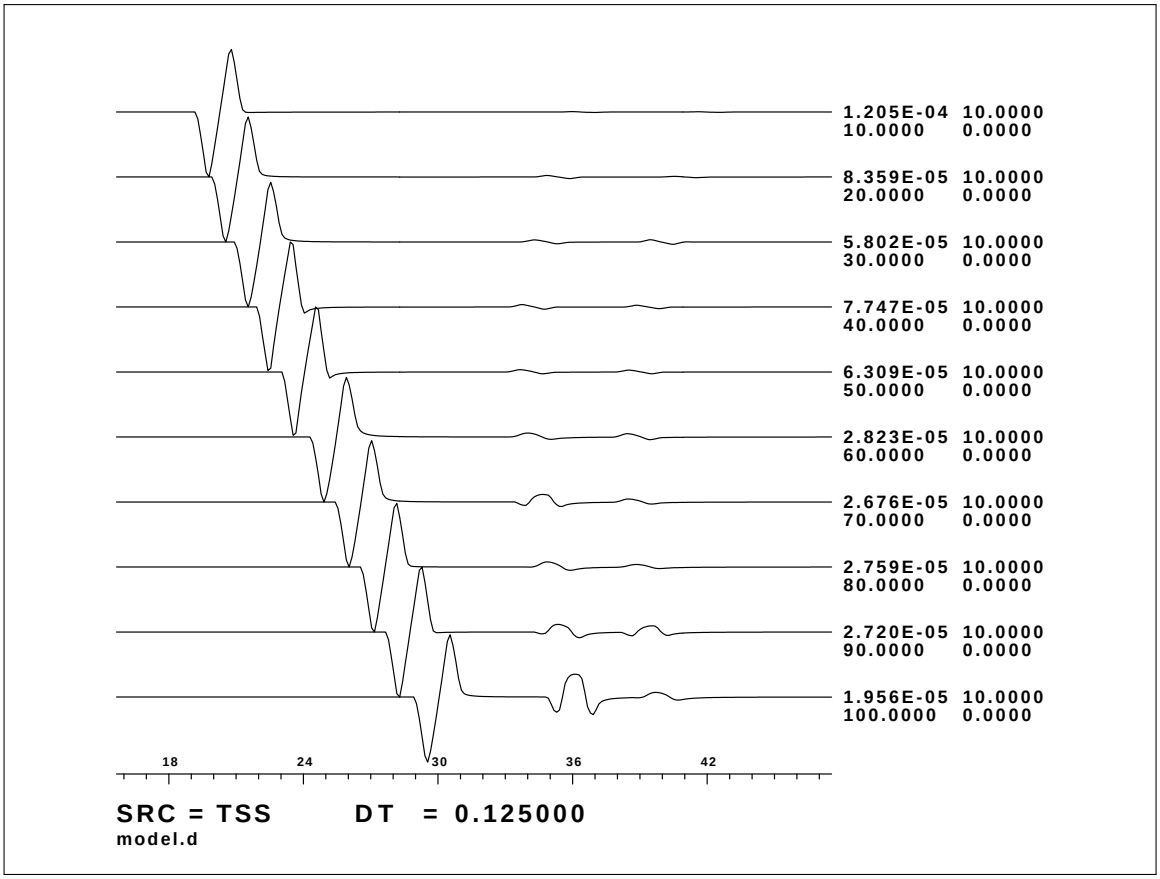
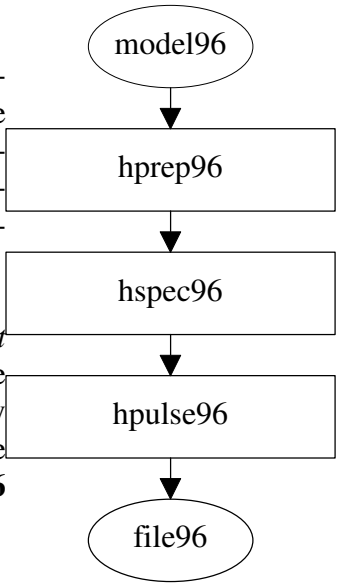


Fig. 6. The TSS time histories file generated by **fprof96**.

CHAPTER 3
WAVENUMBER INTEGRATION REVISED

1. Introduction



This chapter describes the use of wavenumber integration for the generation of synthetic seismograms. This type of seismogram synthesis is complete, but can be computationally intensive. There are three stages between the specification of the model file and the final synthetic seismograms.

The program **hprep96** creates a data file *hspec96.dat* for use by **hspec96** to create the Green's functions in the ω-distance space. The output of this program is a binary file, *hspec96.grn*, which is used by **hpulse96** to convolve the response with the source time function to create **file96** Green's function time histories.

2. hprep96

The purpose of this program is to generate a data file for use by the wavenumber integration programs. These programs are more general than the generalized ray programs in that one may consider more than a single distance, source depth or receiver depth. This permits creating record sections for vertical seismic profiling as well as ordinary refraction lines. However there is the requirement to compute synthetics using the same number of points, N, and the sample time domain sampling interval Δt.

To generate synthetics as free as possible from numerical artifacts, due to the time and space domain periodicity caused by sampling in the frequency - wavenumber domain, certain parameters must be carefully defined. These are α and L. α is used to alleviate the time domain periodicity, and should be such that $\alpha N \Delta t \approx 2.5$, which usually works well unless the layer multiples do not decay rapidly. This default choice ensures that arrivals that wrap around to interfere with the desired signal are reduced by a factor of $e^{2.5}$. The wavenumber sampling is $\Delta k = 2\pi / L$, where the L parameter is defined using the criteria developed by Bouchon.

This program will read a model, the distance file, and get source and receiver depth information from the command line, and will generate the data file required by the

wavenumber integration programs.

Unless overridden by specific command line arguments, the program will attempt to define suitable values of α and L.

Figure 1 shows the processing flow for this program. The program requires one earth model file in the *model96* format and two optional control files. The output consists of the file *hspec96.dat*.

hprep96 *[flags]*, where the command flags are

-M model (default none, this is required) Name of earth model file.

-d dfile (default none, this is required) Name of distance file
The required distance file. The file contains the following ASCII entries per line:

DIST DT NPTS T0 VRED

where **DIST** is the epicentral distance in kilometers, **DT** is the sampling interval for the time series, **NPTS** is the number of points in the time series (a power of 2). **T0** and **VRED** are used to define the time of the first sample point which is **T0 + DIST/VRED** if **VRED** $\neq$ **0** or **T0** if **VRED = 0**.

-FHS srcdep (overrides -HS) Name of source depth file

-FHR recdep (overrides -HR) Name of receiver depth file

-HS hs (default 0.0) Source depth

-HR hr (default 0.0) Receiver depth

-TF (default true) top surface is free

-TR (default false) top surface is rigid

-TH (default false) top surface is halfspace

-BF (default false) bottom surface is free

-BR (default false) bottom surface is rigid

-BH (default true) bottom surface is halfspace

-ALL (default true) Compute all Green's functions

-EQEX (default false) Compute earthquake/explosion Green's functions

-EXF (default false) Compute explosion/point force Green's functions

-CMAX cmax -C1 c1 -C2 c2 -CMIN cmin (default none) phase velocity filter band

-XL xleng (default automatic determination) $\Delta\mathbf{k} = \mathbf{6.2831853/xleng}$

-XF xfac (default 4.0) Upper bound in wvno integration parameter at a given a given frequency is $\mathbf{k} = \mathbf{xfac}\, 2\pi\mathbf{f/v_{min}}$.

-NDEC ndec (default 1) decimate the time series

-ALP alp (default 2.5) time domain damping factor. The end of the trace is reduced by a factor of $\mathbf{e^{-alp}}$ to reduce the effects of Discrete Fourier Transform periodicity and to remove poles from the real wavenumber axis.

-Z The first time point will be t0+abs(source depth - receiver depth)/vred

-R the first time point will be t0+sqrt(z*z + r*r)/vred

-V (default false) Force verbose output

-?

-h (default none) provides assistance on command usage

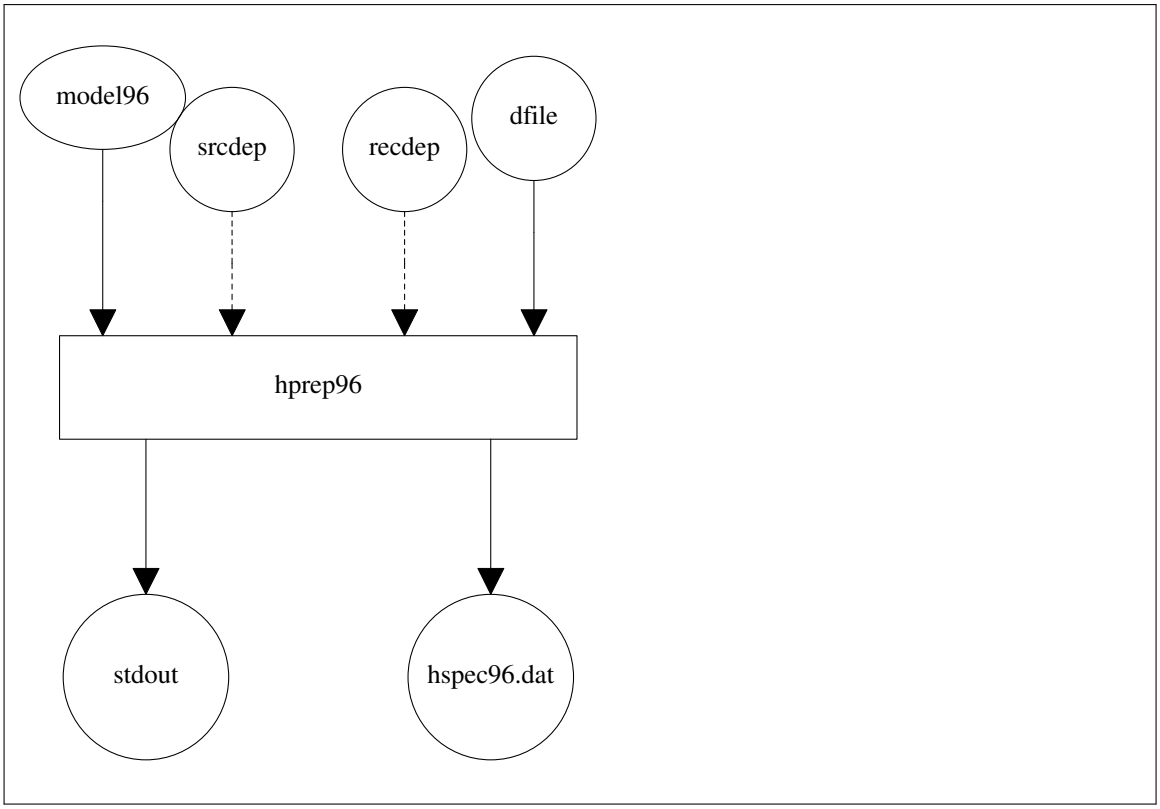


Fig. 1. Processing flow for **hprep96**

Of the command line parameters, the **NDEC -ndec** flag is very useful in ensuring that the **hspec96.dat** file is correct for a given problem. The computational effort is **O(f^2)**, which means that if the number of frequencies is doubled, e.g., by doubling **NPTS**, the computation time increases at least by a factor of 4. The only way to determine if the xleng parameter, **L**, is too small is to create a record section. Numerical noise with negative moveout will appear in this case. The noise will also be apparent at low frequencies. When starting work with a new model, it is prudent to make a simple record section, but to save computer time, decrease the number of points in the time series by **ndec** keeping **N**Δ**t** fixed. One would invoke the processing sequence with a suitably large value of **ndec**, examine the results, and then repeat the computations with **ndec** = **1** for the final run.

3. hspec96

Figure 3 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

hspec96 *[flags]*, where the command flags are

-H (default false) Use Hankel function not Bessel. This will be useful at large distances and high frequencies, especially when phase velocity windowing is

performed.

-A arg (default arg=3.0) value of **kr** where $H_n(kr)$ replaces $J_n(kr)$ in integration - only used when -H is used

-K (default Futterman) use Kjartansson Causal Q

The following govern wavefield at source. The default is the entire wavefield

-SU (default whole wavefield) Compute only upgoing wavefield from the source

-SD (default whole wavefield) Compute only downgoing wavefield from the source

-SPUP Include upward P at source

-SSUP Include upward S at source

-SPDN Include downward P at source

-SSDN Include downward S at source

The following govern the wavefield at the receiver. The default is the entire wavefield. The usefulness and effect of these for a surface receiver is not clear.

-RD Include only downgoing waves at receiver

-RU Include only upgoing waves at receiver

-RPUP Include upward P at receiver

-RSUP Include upward S at receiver

-RPDN Include downward P at receiver

-RSDN Include downward S at receiver

-?

-h Online help concerning program usage

4. hpulse96

This was updated June, 2021. The command line options are more specific about the resulting file and the output units.

Figure 3 shows the processing flow for this program. The program requires the *hspec96.grn* file created by **hspec96(V)** and optionally the source pulse definition file *rfile*. The program output is on *stdout* and is a time series in **file96(V)** format.

Program control is through the command line. The online help concerning program usage is

USAGE: **hpulse96 [-t -o -p -i] -a alpha -l L [-D -V -A] [-F rfile] [-m mult] [-STEP|-IMP] [-Z] [-?] [-h]**

Output time series in ASCII file96 format

TIME FUNCTION SPECIFICATION

-t Triangular pulse of base 2 L dt

-p Parabolic Pulse of base 4 L dt

-l L (default 1)duration control parameter

-o Ohnaka pulse with parameter alpha

-i Dirac Delta function

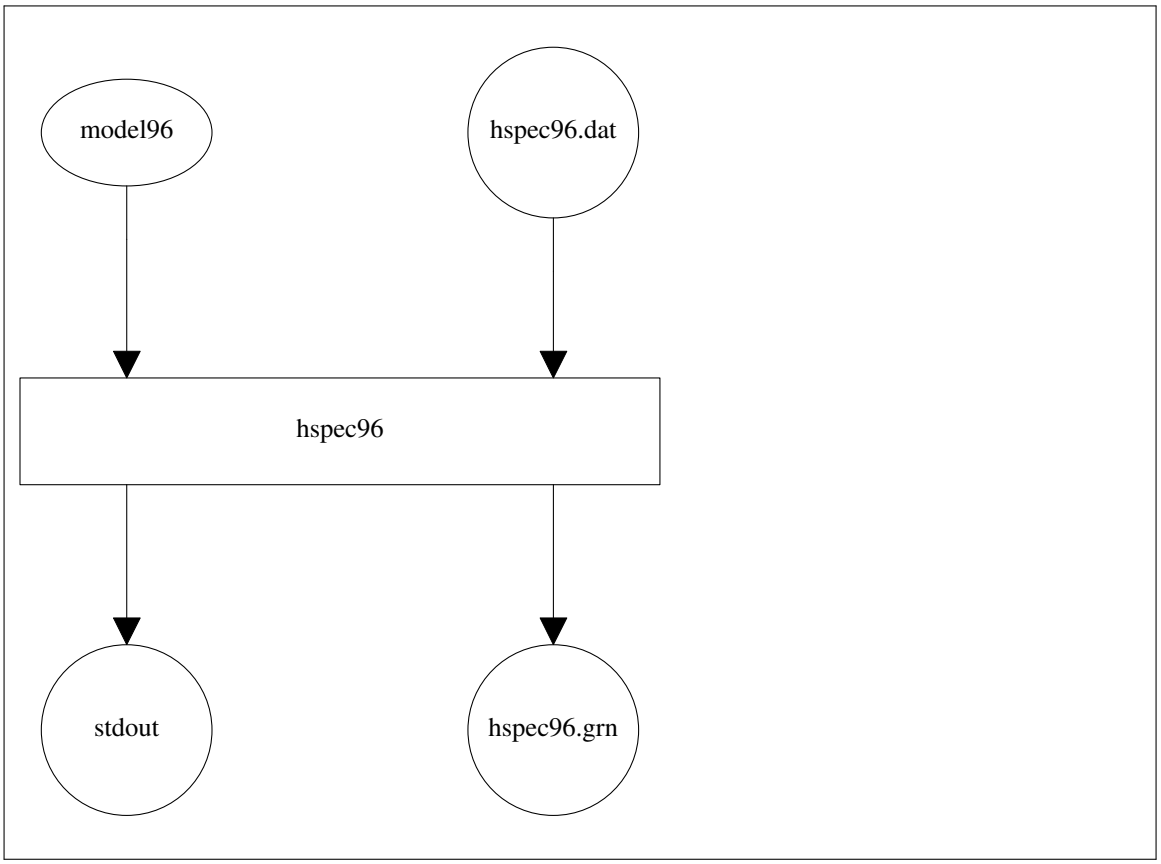


Fig. 2. Processing flow for **hspec96**

-a alpha Shape parameter for Ohnaka pulse
-F rfile User supplied pulse
-m mult Multiplier (default 1.0)
-Z (default false) zero phase triangular/parabolic pulse
By default the source time function is
-STEP (default) steplike integral of aboce pulses
-IMP impulse like pulse with unit area
steplike. -IMP forces impulse like. -D -IMP is Green s function
These do not define the shape but rather the
shape of the source pulse. For earthquake
studies use the default steplike

OUTPUT and UNITS
-D Output is ground displacment
-V Output is ground velocity (default)
-A Output is ground acceleration
If the model is km, km/s, gm/cm^3 then the output is
Option units
-A cm/s/s for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne
-V cm/s for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne

-D cm for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne
In a fluid the stress is in Pa for a
moment of 1.0e+16 dyne-cm or force 1.0e+14 dyne
If the model is MKS, e.g, m, m/s, kg/mˆ3, then
ZRT Greens functions are for moment 1.0 N-m, force
1.0 N with units m, m/s, m/s/s. The P stress are in Pa

-? Write this help message
-h Write this help message

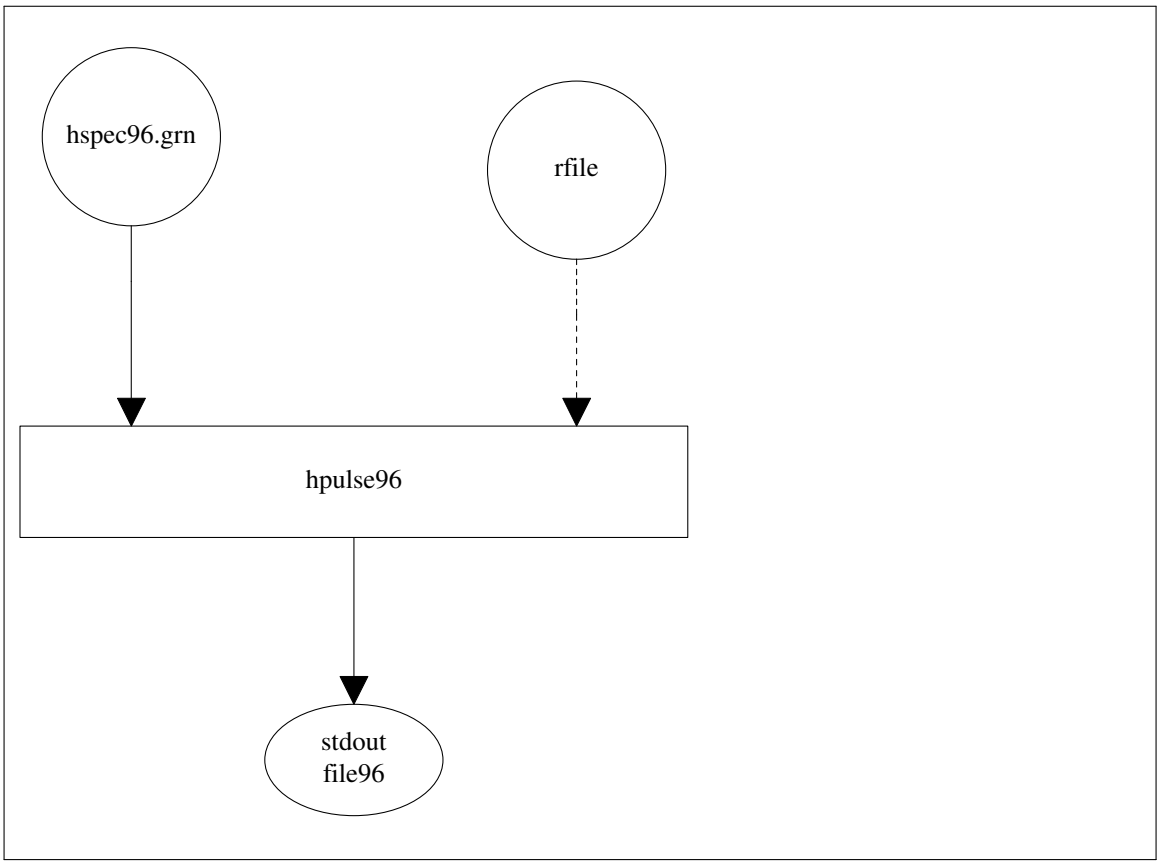


Fig. 3. Processing flow for **hpulse96**

5. Sample Run

Given the sample model and distance files of Chapter 1, the following commands are run:

```
rbh> hprep96  -M model.d -d dfile -HS 10 -HR 0 -ALL

rbh> hspec96   > hspec96.out

rbh> hpulse96 -V -p -l 4  | fprof96
```

The synthetic transverse time histories for a vertical strike-slip source are given in Figure 4.

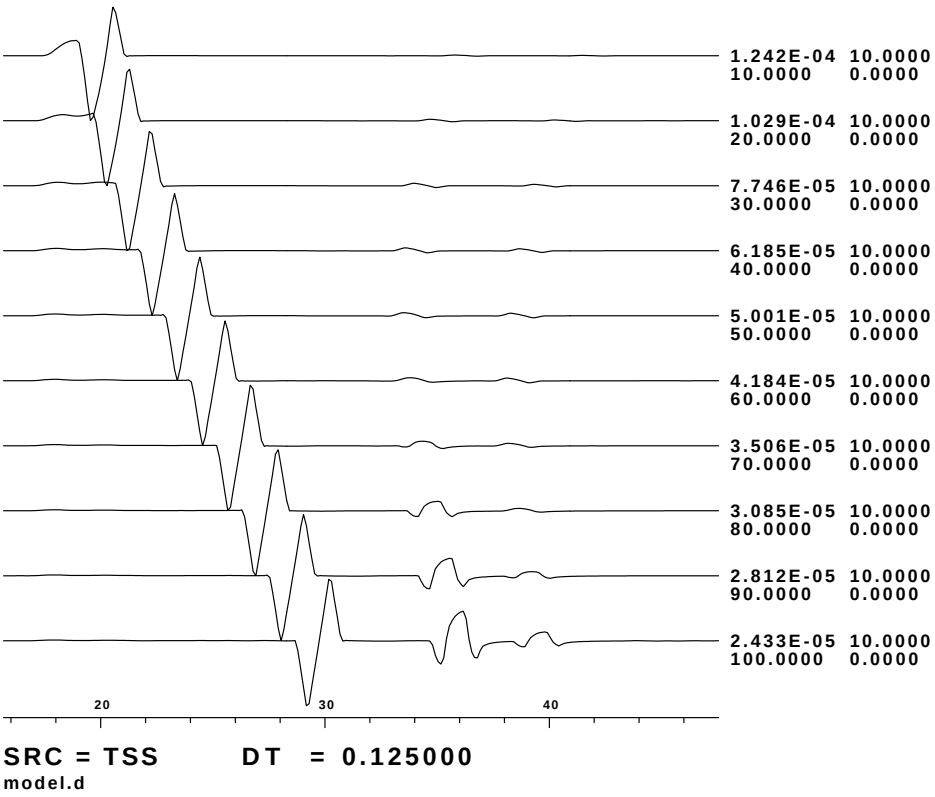


Fig. 4. TSS Green's function for the layered model.

6. hwhole96

This program computes synthetics for a wholespace using the properties of the first layer of the model. This program has two important uses. First, a wholespace solution is fundamental in the development of seismic wave theory. Second, this analytical solution permits a direct test of the wavenumber integration used by **hspec96**. Since an analytical solution is used, wavenumber integration is not performed. The excitation of this program is very rapid. Figure 5 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and

on a binary file **hspec96.grn**.

Program control is through the command line:

hwhole96 *[flags]*, where the command flags are

-K (default Futterman) use Kjartansson Causal Q

-?

-h

Online help concerning program usage

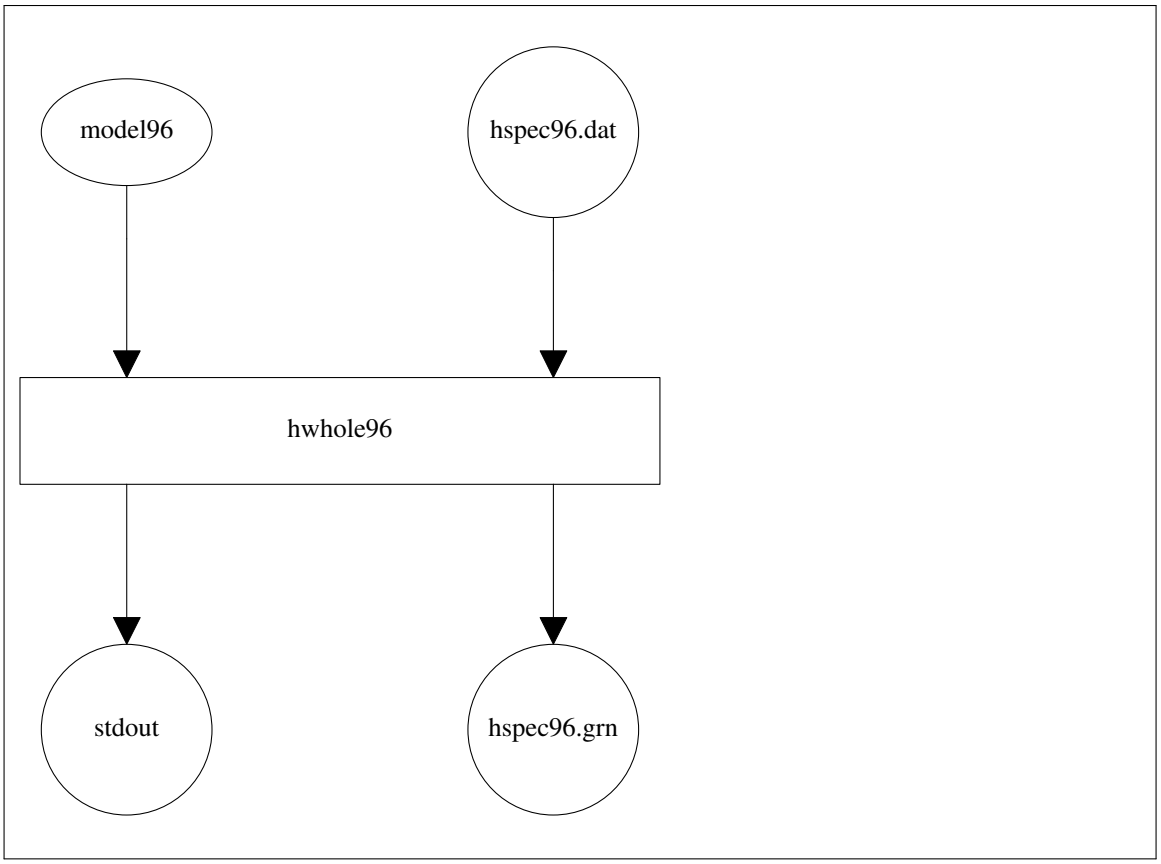


Fig. 5. Processing flow for **hwhole96**

```
rbh> hprep96  -M model.d -d dfile -HS 10 -HR 0 -ALL

rbh> hwhole96   > hspec96.out

rbh> hpulse96 -V -p -l 4  | fprof96
```

The synthetic transverse time histories for a vertical strike-slip source are given in Figure 6.

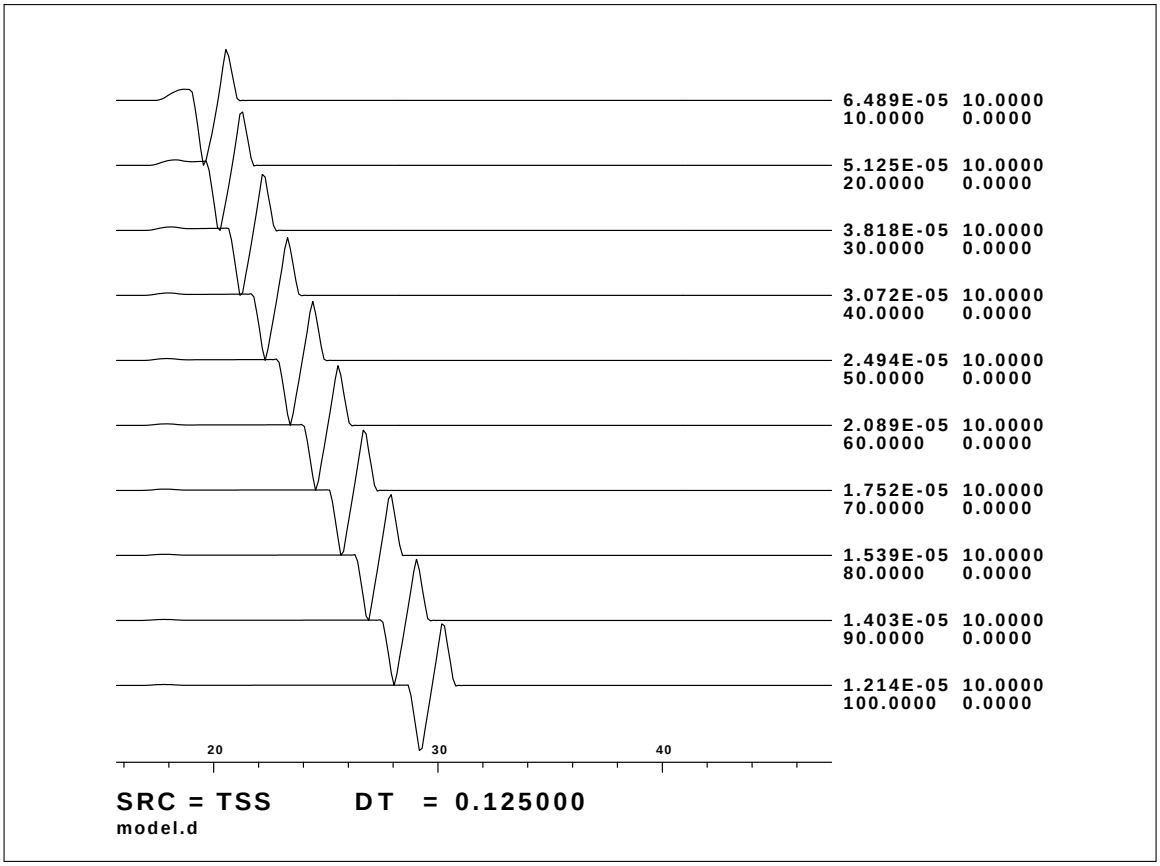


Fig. 6. Whole space response for TSS Green's function.

7. hspec96p

This program computes the p-τ response for a layered media. The wavenumber integration method for computing synthetic seismograms consists of evaluating the double integral

$$g(r,t) = \int_{-\infty}^{\infty} g(r,f)e^{j2\pi ft}df$$

where

$$g(r,f) = \int_{0}^{\infty} g(k,f)J_n(kr)kdk$$

Since this equation is a Hankel transform, we also have

$$g(k,f) = \int_{0}^{\infty} g(r,f)J_n(kr)rdr$$

In exploration seismology it convenient to think in terms of ray parameter p which is related to k and $\omega = 2\pi f$ by $k = p\omega$. By substitution, we can define $g(p,\tau)$ as

$$\mathbf{g(p,\tau) = g(p,t-pr) = \int_{-\infty}^{\infty} g(k = 2\pi fp, f) e^{j2\pi f(t-pr)} df}$$

$$= \int_{-\infty}^{\infty} g(k = 2\pi fp, f) e^{j2\pi f\tau} df$$

This last expression shows that the $p-\tau$ response is actually the inverse Fourier transform of g(k, f) with the constraint $k = p2\pi f$.

The $p-\tau$ seismogram has several interesting properties. Reflection hyperbolas in the r – t domain, appear as ellipses in the $p-\tau$ domain. A refracted arrival with ray parameter, p_{refr}, is mapped into a point with ray parameter p_{refr} in the $p-\tau$ domain. More importantly, the effect of the Hankel transform is that geometrical spreading is removed. The $p-\tau$ time history gives correct relative amplitudes due to plane wave reflection and refraction in the model.

Figure 7 shows the processing flow for this program. The program requires the *hspec96p.dat* file created by **hprep96p(VI)**. The program output is on *stdout* and on a binary file **hspec96.grn**.

Program control for the program **hprep96p** is through command line flags. The program is similar in concept to that of **hprep96**, except for two important points. First the output is a file by the name *hspec96p.dat*. Since we are interested in p-τ time histories, distance is immaterial; ray parameter is the important parameter. The distance file, indicated by the command line flag **-d** *dfile* is important since this defines the sampling interval and the number of points in the time series. The use of the same file rather than another one, means that one can easily obtain time histories in the r-t space and the p-τ space using the same data files, as indicated by the examples in this chapter. The program **hprep96p** is executed as follows:

`hprep96p` *[flags]*, where the command flags are

`-M model` (default none) Earth model file

`-d dfile` (default none) Name of distance file

The required distance file. The file contains the following ASCII entries per line:

DIST DT NPTS T0 VRED

where **`DIST`** is the epicentral distance in kilometers, **`DT`** is the sampling interval for the time series, **`NPTS`** is the number of points in the time series (a power of 2). *For this program the entries DIST T0 and VRED are NOT used.* They are include in the input for compatibility with the synthetic seismogram programs.

`-FHS srcdep` (overrides -HS) Name of source depth file

`-FHR recdep` (overrides -HR) Name of receiver depth file

`-HS hs` (default 0.0) Source depth

`-HR hr` (default 0.0) Receiver depth

`-TF` (default true) top surface is free

`-TR` (default false) top surface is rigid

`-TH` (default false) top surface is halfspace

-BF (default false) bottom surface is free
-BR (default false) bottom surface is rigid
-BH (default true) bottom surface is halfspace
-ALL (default true) Compute all Green s functions
-EQEX (default false) Compute earthquake/explosion Green s functions
-EQF (default false) Compute explosion/point force Green s functions
-PMIN pmin -PMAX pmax -DP dp (default none) ray parameter sample space in *sec/km*
-TRUE (default false) use modified p-tau response. For a simple direct body wave arrival, the pulse of the vertical and horizontal components are Hilbert transforms of each other, because has a different order **n** of the Bessel functions. This program automatically adjusts the phase of the radial components by π **/ 2** radians, unless specifically instructed not to by this command. Application of the correct order of the Hankel transform to observed three component data would result in the same phase difference, and this option is provided for a direct comparison.
-NDEC ndec (default 1) decimate the time series
-ALP alp (default 2.5) time domain damping factor. The end of the trace is reduced by a factor of $\mathbf{e}^{-\mathbf{alp}}$ to reduce the effects of Discrete Fourier Transform periodicity. Note, if **hspec96p** is used for site spectral response, then set **alpha = 0.0**. For time series, use the default.
-? (default none) this help message
-h (default none) this help message

Program control for **hspec96p** is through the command line:

hspec96p *[flags]*, where the command flags are
-K (default Futterman) use Kjartansson Causal Q
The following govern wavefield at source. The default is the entire wavefield
-SU (default whole wavefield) Compute only upgoing wavefield from the source
-SD (default whole wavefield) Compute only downgoing wavefield from the source
-SPUP
Include upward P at source
-SSUP
Include upward S at source
-SPDN
Include downward P at source
-SSDN
Include downward S at source
The following govern wavefield at receiver. The default is the entire wavefield
-RD Include only downgoing waves at receiver
-RU Include only upgoing waves at receiver
-RPUP Include upward P at receiver
-RSUP Include upward S at receiver
-RPDN Include downward P at receiver
-RSDN Include downward S at receiver
-?
-h

Online help concerning program usage

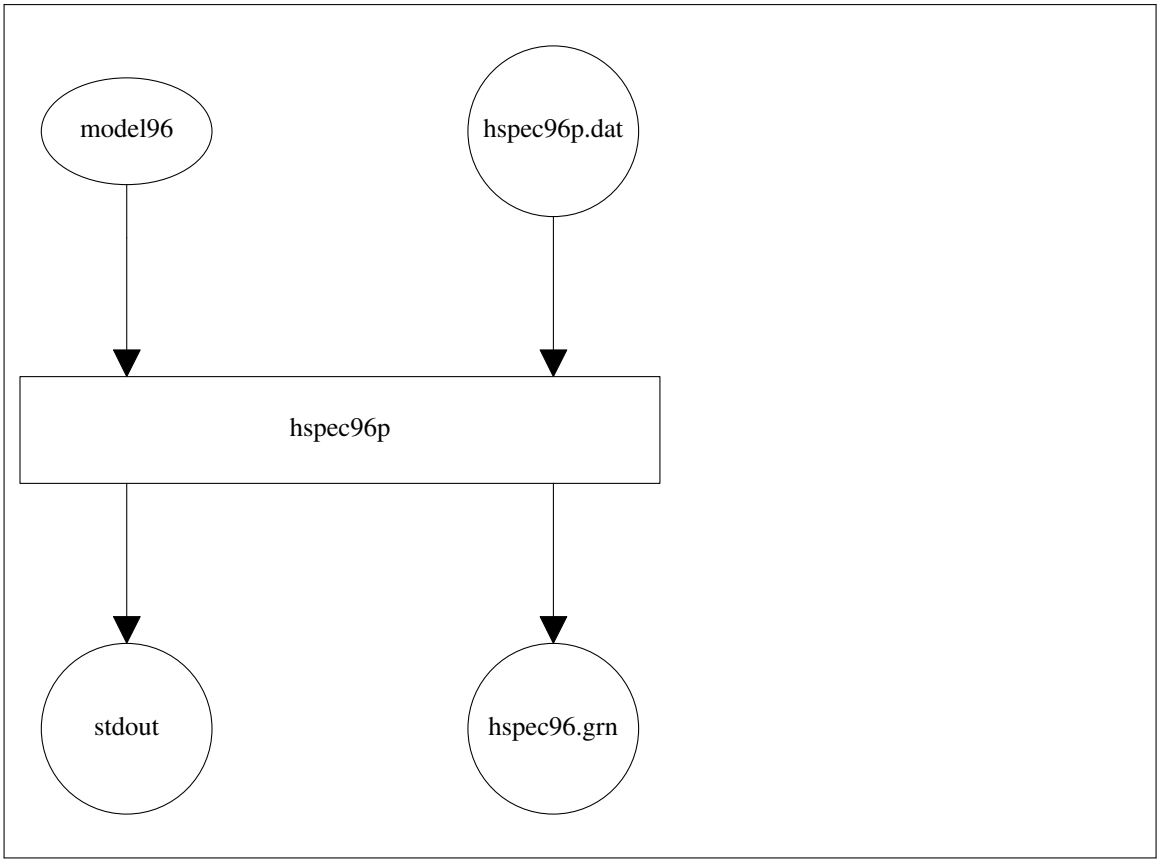


Fig. 7. Processing flow for **hspec96p**

```
rbh> hprep96p -M model.d -d dfile -HS 10 -HR 0 -ALL \
          -PMIN 0.01 -PMAX 0.20 -DP 0.01

rbh> hspec96p   > hspec96p.out

rbh> hpulse96 -V -p -l 4  | fprof96
```

The synthetic transverse time histories for a vertical strike-slip source are given in Figure 8. Note how the reflected amplitudes are enhanced, since geometrical spreading has been removed.

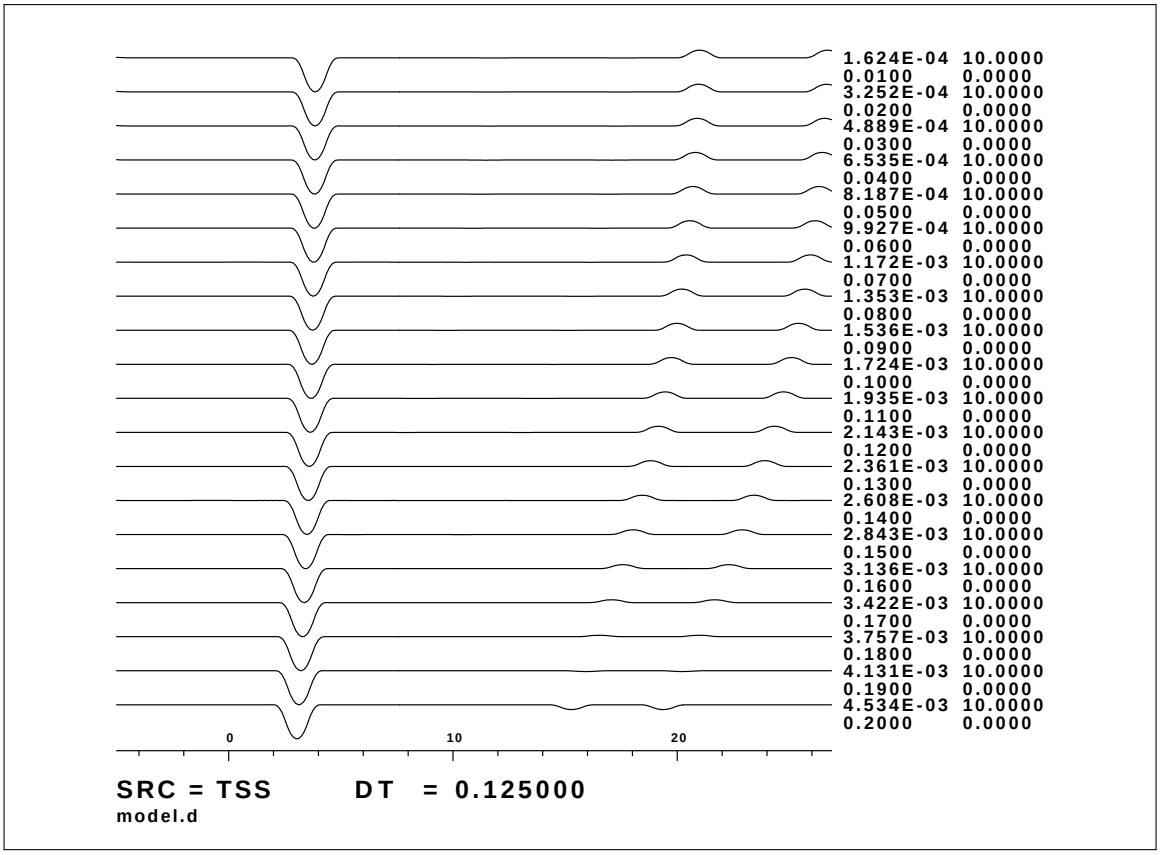


Fig. 8. The p-τ response for the TSS Green's function. The annotation at the right indicated the maximum trace amplitude, the fact that the source depth was 10.0 km, the ray parameter, and the receiver depth.

8. rspec96 and rspec96p NEW

The programs **hspec96** and **hspec96p** were designed for crustal earthquake studies. The code was extended to permit a model with all fluid layers and models with a stack of fluid layers at the top or the bottom. The implementation does not permit a model with fluid layers embedded in a solid structure. The reason for the problem is that propagator matrices were used. One way to address such models with **hspec96** and **hspec96p** would be to approximate the fluids with a solid having a very low S-wave velocity. But to do this, it is necessary to compare the results to a formulation that permits such a medium.

The **rspec96** and **rspec96p** programs are based on the generalized reflection matrix technique of Chen (1993) with modification by Pei et al (2008,2009). Extension to fluid layers was done by Wu and Chen (2016).

The programs are run in the same manner as **hspec96**. The comparison for executing

the codes is

hspec96	rspec96
hprep96 -M model -d dfile	hprep96 -M model -d dfile
hspec96	rpsec96
hpulse96 -V -p -l	hpulse96 -V -p -l 1

As with **hspec96** and **hspec96p**, these codes only work with isotropic models. There has been no extension to transverse isotropy as was done with **tspec96** and **tspec96p**.

As an example of running these programs, consider the following shell script:

```
#!/bin/sh

#####
#  define the two models
#####

cat > S.mod << EOF
MODEL.01
Fluid layer model
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM) VP(KM/S) VS(KM/S) RHO(GM/CC)   QP   QS  ETAP  ETAS  FREFP  FREFS
  4.0000  6.0000  3.5000  2.7000 0 0 0 0 1 1
  4.0000  6.0000  0.0000  2.7000 0 0 0 0 1 1
  7.0000  6.0000  3.5000  2.7000 0 0 0 0 1 1
 26.0000  6.0000  3.5000  2.7000 0 0 0 0 1 1
   .0000  8.0000  4.7000  3.3000 0 0 0 0 1 1
EOF

cat > Sa.mod << EOF
MODEL.01
Approximate fluid layer with low S velocity
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM) VP(KM/S) VS(KM/S) RHO(GM/CC)   QP   QS  ETAP  ETAS  FREFP  FREFS
  4.0000  6.0000  3.5000  2.7000 0 0 0 0 1 1
```

```
  4.0000   6.0000   0.0100   2.7000 0 0 0 0 1 1
  7.0000   6.0000   3.5000   2.7000 0 0 0 0 1 1
 26.0000   6.0000   3.5000   2.7000 0 0 0 0 1 1
   .0000   8.0000   4.7000   3.3000 0 0 0 0 1 1
EOF

#####
#  define source depth
#####
HS=20

#####
#  define receiver depths
#####
cat > FHR << EOF
3.6
3.8
4.0
4.2
4.4
4.6
4.8
5.0
5.2
5.4
5.6
5.8
6.0
6.2
6.4
6.6
6.8
7.0
7.2
7.4
7.6
7.8
8.0
8.2
8.4
EOF

P=0.05
DT=0.25
cat > dfile << EOF
100.0 ${DT} 512 0 0
EOF
BC="-TF -BH"

#####
#  make Greens functions, place them in the subdirectory FINAL.DIR
#  rename them with R at end for rspec (fluid layer)  and with
#  a at end for approximation
#####
```

```
rm -fr FINAL.DIR
mkdir FINAL.DIR

MODEL=S.mod
#####
hprep96 -M $MODEL -d dfile -FHR FHR -HS ${HS} ${BC} -ALL  -NDEC 1
rm -f hspec96.grn
time rspec96 > rspec.out 2>&1
rm -f B*.sac
hpulse96 -V -p -l 1 | f96tosac -B
(cd FINAL.DIR
for i in ../B*.sac
do
     STEL=`saclhdr -STEL $i | awk '{printf "%5.2f",$1}' `
     KCMPNM=`saclhdr -KCMPNM $i`
     FNAME=`echo ${KCMPNM} ${STEL} | awk '{printf "%s%sR",$1,$2}' `
     mv $i ${FNAME}
done
)
rm -f hspec96.grn
rm -f B*.sac

#####

MODEL=Sa.mod
hprep96 -M $MODEL -d dfile -FHR FHR -HS ${HS} ${BC} -ALL  -NDEC 1
rm -f hspec96.grn
time hspec96 > hspec.out 2>&1
rm -f B*.sac
hpulse96 -V -p -l 1 | f96tosac -B
(cd FINAL.DIR
for i in ../B*.sac
do
     STEL=`saclhdr -STEL $i | awk '{printf "%5.2f",$1}' `
     KCMPNM=`saclhdr -KCMPNM $i`
     FNAME=`echo ${KCMPNM} ${STEL} | awk '{printf "%s%sa",$1,$2}' `
     mv $i ${FNAME}
done
)
rm -f hspec96.grn
rm -f B*.sac

#####
#  use gsac to plot a record station
#  as s function of receiver depth
#####

for GRN in RDS
do
gsac << EOF
r FINAL.DIR/${GRN}*[Ra]
bg plt
color list blue red
prs sa 0 amp 0.2 stel reverse vl 3.5 8.5
```

```
q
EOF
mv PRS001.PLT a${GRN}.PLT
plotnps -F7 -W10 -EPS -K < a${GRN.PLT} > a${GRN}.eps
done

#####
#  clean up
#####
rm PRS001.CTL
rm hspec.out rspec.out hspec96.dat FHR S.mod Sa.mod dfile
rm -fr FINAL.DIR
```

The comparison to for the RDS Greens function is show in Figure 9. In this true amplitude plot the fluid lays is between depths of 4.0 and 8.0 km. The blue curve is the result of running **rspec96** with the fluid layer and the red curve is the result of running **hspec96** with the low S-wave velocity approximating the fluid layer. The first thing to notice is that the signal incident from the below generates an S wave at the sharp contrast in S velocity at a depth of 8 km. The resultant signal is obvious as the large pusle with the large moveout. On the other hand the signals in the solid layers are virtually the same. The problem of Discrete Fourier Transform wrap-around is reduced the the use of complet frequency (the *-ALP alp* parameter in **hprep96**. We are thus able to assess the limitations of approximating the fluid layer with one having low S-wave velocity.

The other aspect is that **hspec96** takes about one-third longer to execute than **rspec96** because it must sample a greater range of wavenumbers in order to compute the low velocity S-wave arrival.

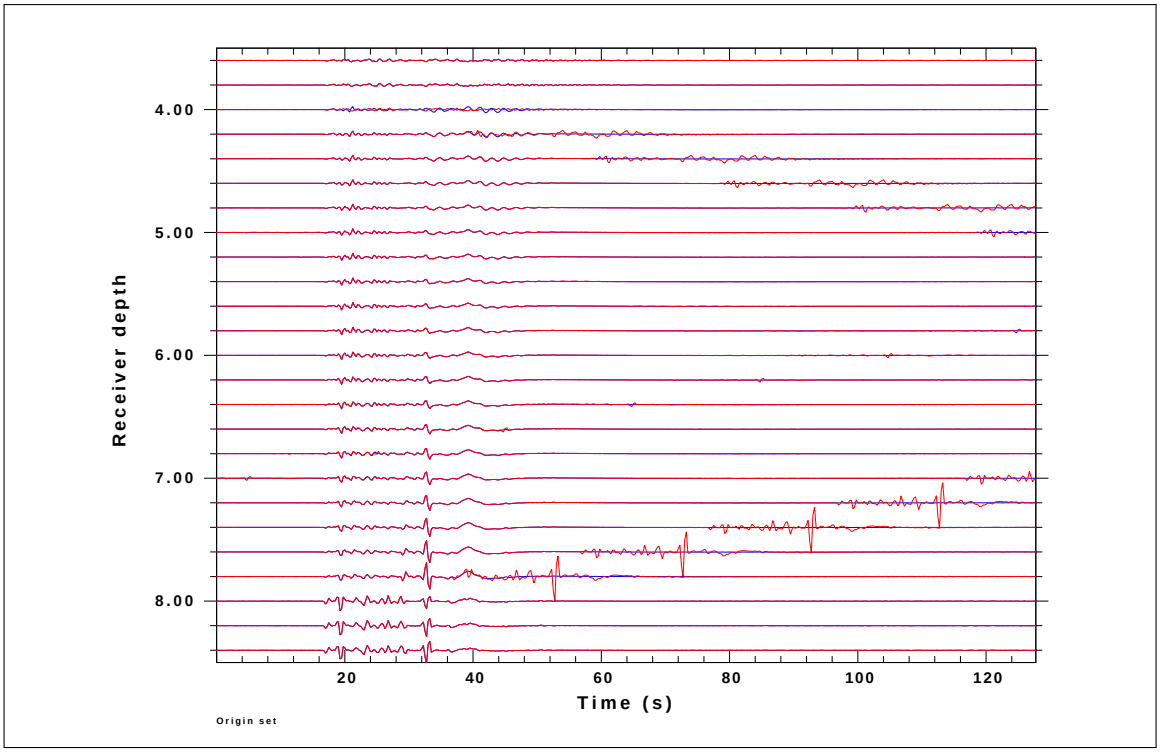


Fig. 9. Plot of the radial component velocities as a funciton of receiver depth. (blue) **rspec96** computation with the fluid model and (red) the **hspec96** computation with the approximate model. The annotation at the right indicated the maximum trace amplitude, the fact that the source depth was 10.0 km, the ray parameter, and the receiver depth.

References

Chen, X. (1993). A systematic and efficient method of computing normal modes for multilayered half-space, *Geophys. J. Int*

Chen, X. (1993). A systematic and efficient method of computing normal modes for multilayered half-space, Geophys. J. Int. **115**, 391-409.

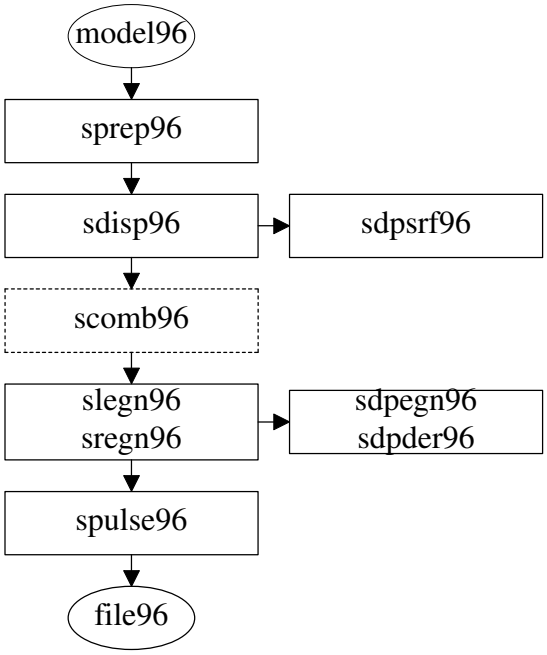
Pei, D., J. N. Louis and S. K. Pullammanappallil (2008). Improvements on computation of phase velocities of Rayleigh waves based on the generalized R/T coefficient method, *Bull. Seism. Soc. Am.* **98**, 200-287.

Pei, D., J. N. Louis and S. K. Pullammanappallil (2009). Erratum to Improvements on computation of phase velocities of Rayleigh waves based on the generalized R/T coefficient method, *Bull. Seism. Soc. Am.* **99**, 2610-2611.

Wu, B. and X. Chen (2016). Stable, accurate and efficient computation of normal modes for horizontal stratified models, *Geophys. J. Int.* **206**, 1201-1300.

CHAPTER 4
MODAL SUMMATION REVISED

1. Introduction



This chapter describes the use of modal superposition of surface waves to create synthetic seismograms. For low frequency seismograms, this technique can be much faster than wavenumber integration and generalized ray techniques. In addition, extension of the model by adding a high velocity cap zone at great depth permits the computation of almost complete synthetic seismograms by the *locked mode* approximation. There are four stages between the specification of the model file and the final synthetic seismograms.

The program **sprep96** creates a data file *sdisp96.dat* for use by **sdisp96** to determine the multimode phase velocity dispersion for the model. **slegn96** and **sregn96** compute the eigenfunctions required for synthetics from the dispersion curves. Finally, **spulse96** uses the eigenfunction and dispersion information to create the required Green's functions in **file96** format. The program **scomb96** can be used to correct the dispersion curves generated by **sdisp96** by filling in missing modes. The programs **sdpsrf96** and **sdpegn96** are used to display the dispersion curves and thus provide a quality control check. The program **sdpder96** lists and plots the depth dependence of surface-wave eigenfunctions and phase velocity partial derivatives.

Given the capabilities of modern computers, it is possible to combine all these functions into a single large program, but the modular approach used permits intervention for quality control at each stage, since determination of dispersion curves is not trivial.

2. sprep96

Figure 1 shows the processing flow for this program. The program requires an earth model file in the *model96* format and two optional control files. The output consists of

the file *sdisp96.dat*.

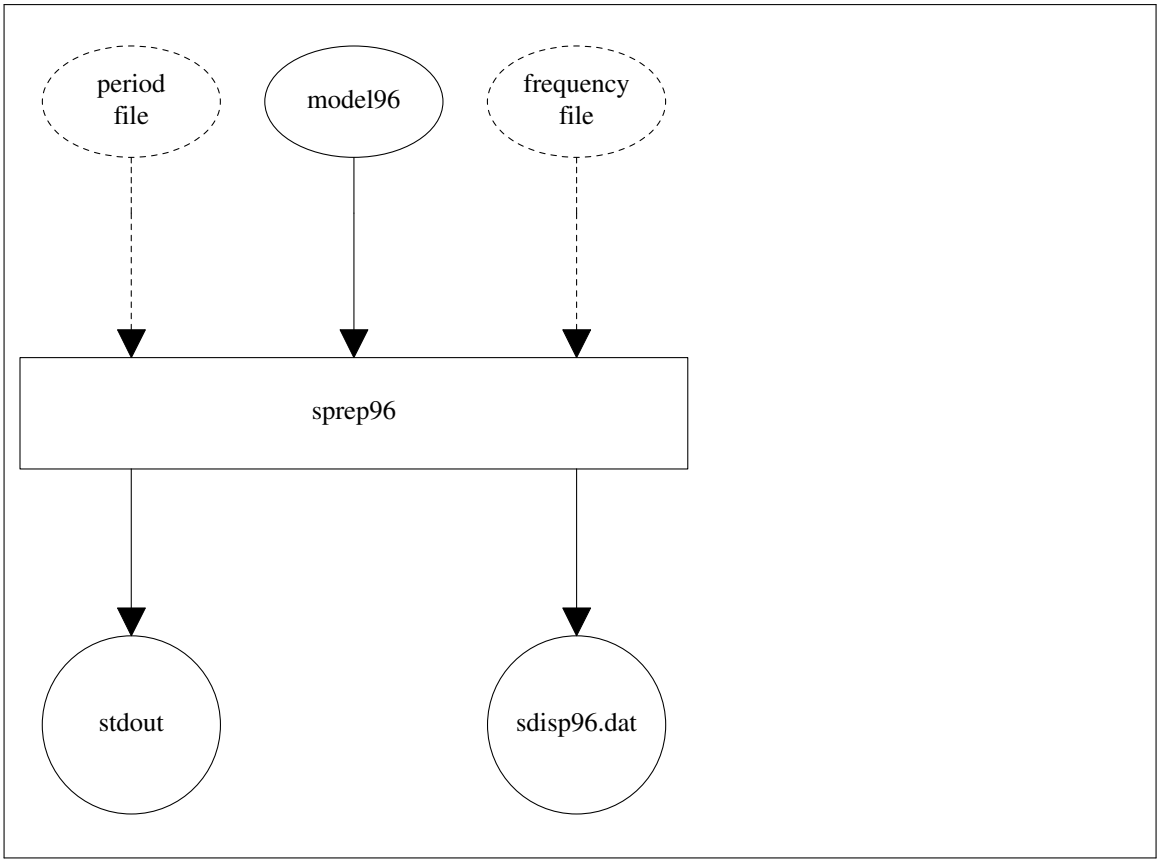


Fig. 1. Processing flow for **sprep96**

Program control is through the command line:

sprep96 *[flags]*, where the command flags are

-M model (default none) Earth model file

-d dfile (default none) Name of distance file

The required distance file. The file contains the following ASCII entries per line:

DIST DT NPTS T0 VRED

where **DIST** is the epicentral distance in kilometers, **DT** is the sampling interval for the time series, **NPTS** is the number of points in the time series (a power of 2). **T0** and **VRED** are used to define the time of the first sample point which is **T0 + DIST/VRED** if **VRED ≠ 0** or **T0** if **VRED = 0**.

This program only uses the DT and NPTS entries to define the frequency range of desired dispersion

-HS hs (default 0.0) Source depth

-HR hr (default 0.0) Receiver depth

-DT dt (default 1.0) Sampling interval

-NPTS npts (default 1) Number of points

-NMOD nmodes (default 1) Maximum number of modes
-L (default false) Generate Love Waves
-R (default false) Generate Rayleigh Waves
Note that at least one of these must be specified. Also note that Love waves do not exist in a halfspace.
-FACL faclov (default 5.0) parameter for controlling root search. A small number is faster, but higher modes may be missed.
-FACR facray (default 5.0) parameter for controlling root search
-FREQ freq User specified single frequency. This permits dispersion computation for a single frequency. This cannot be used for synthetics.
-PER period User specified single period. This permits dispersion computation for a single period. This cannot be used for synthetics.
-FARR freq_file User specified file of separate frequencies on each line. This permits dispersion computation for a list of specified frequencies. This cannot be used for synthetics.
-PARR period_file File of user specified periods User specified file of separate periods on each line. This permits dispersion computation for a list of specified periods. This cannot be used for synthetics.

NOTE: one of -DT -NPTS, -FARR, -PARR, -PER or -FREQ required
-?
-h (default none) this help message

3. sdisp96

This program computes the dispersion curves for the given earth model. This program calculates the desired dispersion curves for the model. All program control is through the file *sdisp96.dat* created by the program **sprep96**. The output consists the of binary dispersion files for Love and Rayleigh waves: *sdisp96.lov* and *sdisp96.ray*, respectively.

Program control is through the command line:

sdisp96 *[flags]*, where the command flags are
-v (default none) On a PC, the frequency of each computation is written to the screen on the same line to indicate that the program is actually working. This cannot be easily done with UNIX since UNIX does not support the formatting parameter '+', but instead scrolls the screen..
-cmin cmrn Instead of starting at about 0.9 Vs(min) start at this velocity
-cmax cmax Instead of ending at Vs(max) start at this velocity
-?
-h (default none) this help message

Figure 2 shows the processing flow for this program.

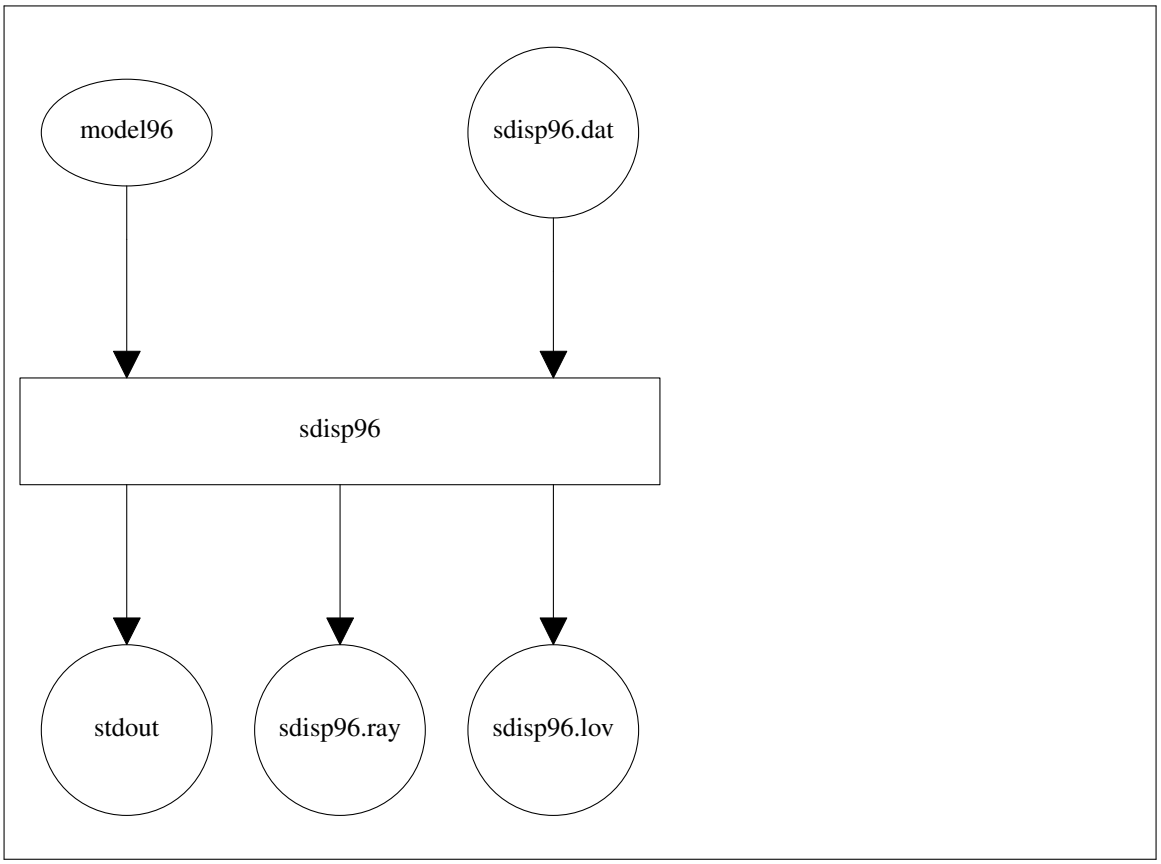


Fig. 2. Processing flow for **sdisp96**

4. scomb96

This program redoes the root determination of *sdisp96* in a user defined rectangular region. This may be necessary if modes are missed. Missing modes are seen by visual inspection of the dispersion curves using **sdpsrf96**. This program calculates the desired dispersion curves for the model. All program control is through the file *sdisp96.dat* and the command line. For Love waves, the input is the file *sdisp96.lov* and the output is the binary dispersion file *tsdisp96.lov*. For Rayleigh waves, the input is the file *sdisp96.ray* and the output is the binary dispersion file *tsdisp96.ray*.

the difference in the input and output file names is to preserve the original file. If many sequential repairs must be made, the user must systematically rename *tsdisp96.lov* to *sdisp96.lov* for Love waves and similarly for the Rayleigh waves.

Program control is through the command line:

`scombf96` *[flags]*, where the command line flags are

`-L` Recompute the Love wave dispersion contained in the file *sdisp96.lov*.

`-R` Recompute the Rayleigh wave dispersion contained in the file *sdisp96.ray*. br *Note that either* **`-L`** *or* **`-R`** *must be specified*

`-XMIN` Minimum value of X-axis

-XMAX Maximum value of X-axis
-CMIN Minimum value of phase velocity for search region
-CMAX Maximum value of phase velocity for search region
-FAC factor Factor to control density of search
-I Ignore existing zeros within [CMIN, CMAX] and restart the search
-?
-h On line command help On line command help

Figure 3 shows the processing flow for this program.

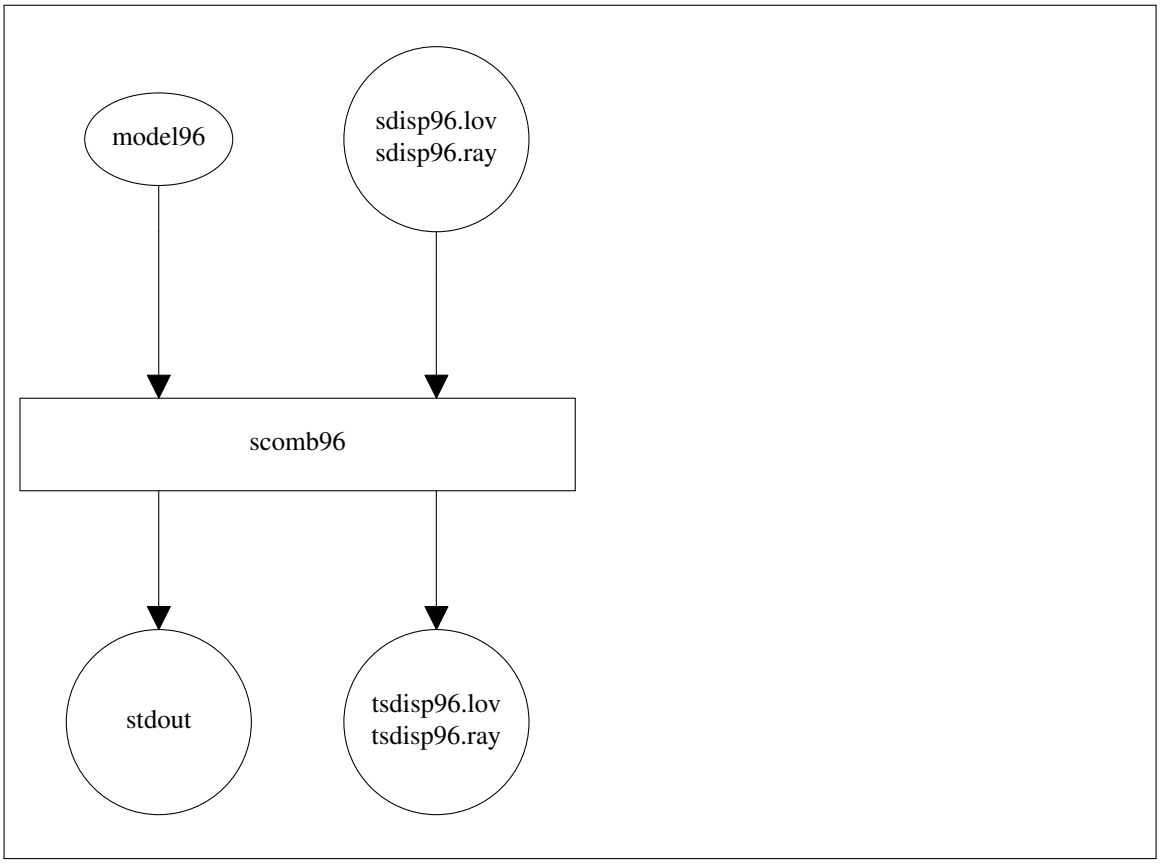


Fig. 3. Processing flow for **scomb96**

5. slegn96/sregn96

Figure 4 shows the processing flow for these programs. The program **slegn96** requires the *sdisp96.lov* file created by **sdisp96** (or the *tsdisp96.lov* file created by **scomb96**) and the program **sregn96** requires the *sdisp96.ray* file created by **sdisp96** (or the *tsdisp96.lov* file created by **scomb96**). The common purpose of these programs is to compute the eigenfunctions corresponding to the dispersion curves for Love and Rayleigh waves, respectively. The eigenfunctions so created for use by the synthetic seismogram program **spulse96** are stored in the files *slegn96.egn* and *sregn96.egn*, respectively.

Program control is through the command line and is identical for each program:

slegn96 *[flags]*, where the command line flags are
-FHS srcdepth_file (overrides -HS) Name of source depth file
-FHR recdepth_file (overrides -HR) Name of receiver depth file
-HS src_depth Source depth, overrides previously given value
-HR rec_depth Receiver depth, overrides previously given value
-NOQ Ignore the Q model. Compute purely elastic dispersion
-DER output all depth dependent values (default false)
-DE output eigenfunctions(depth) (default false)
-DH output DC/DH(depth) (default false)
-DB output DC/DB(depth) (default false)
-DR output DC/DR(depth)
-V (default false) list energy integrals
-?
-h Write this help message (default false)

sregn96 *[flags]*, where the command line flags are
-FHS srcdepth_file (overrides -HS) Name of source depth file
-FHR recdepth_file (overrides -HR) Name of receiver depth file
-HS src_depth Source depth, overrides previously given value
-HR rec_depth Receiver depth, overrides previously given value
-DER output all depth dependent values (default false)
-DE output eigenfunctions(depth) (default false)
-DH output DC/DH(depth) (default false)
-DA output DC/DA(depth). This only applies to the program **sregn96** since Love waves are independent of P-wave velocity. (default false)
-DB output DC/DB(depth) (default false)
-DR output DC/DR(depth)
-V (default false) list energy integrals
-?
-h Write this help message

Note that if the flags **-DER**, **-DH**, **-DA**, **-DB** *or* **-DR** *are set, only the file* **slegn96.der** *or* **sregn96.der** *is computed. If these flags are not set, then only the file* **sregn96.egn** *or* **slegn96.egn** *are created. The files* **slegn96.der** *and* **sregn96.der** *are created for use in waveform inversion or other programs.*

6. spulse96

This was updated June, 2021. The command line options are more specific about the resulting file and the output units.

Figure 5 shows the processing flow for this program. The program requires either or both of the files *slegn96.egn sregn96.egn* to compute synthetic seismograms on the standard output in **file96** format.

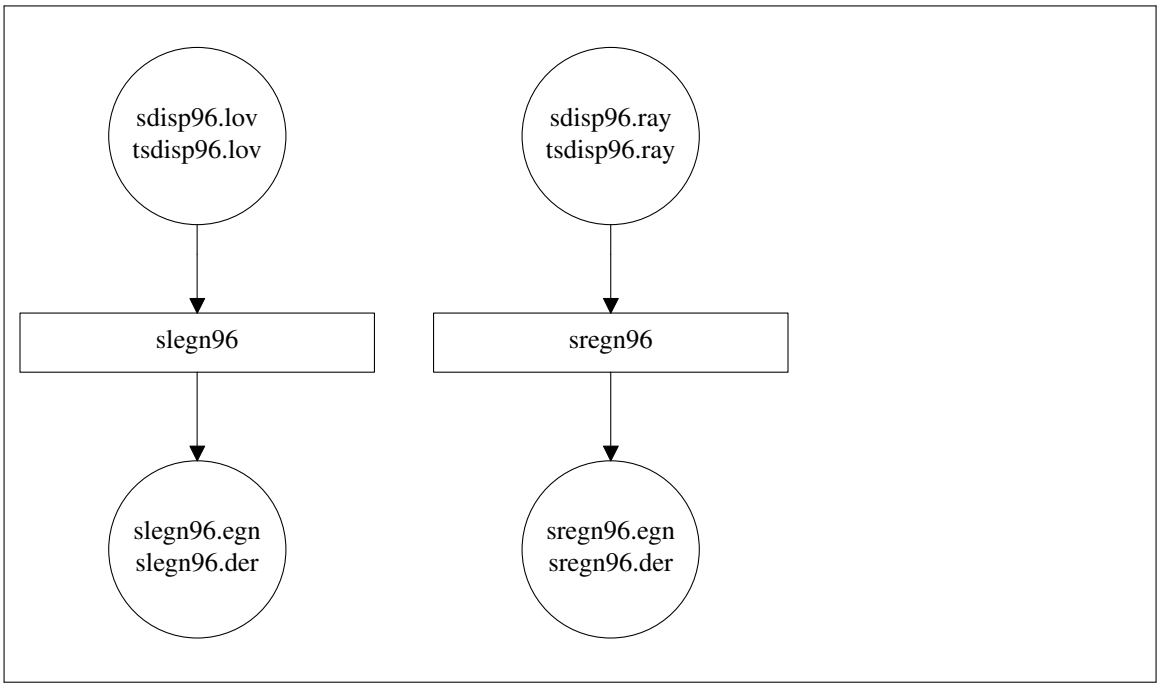


Fig. 4. Processing flow for **slegn96** and **sregn96**

Program control is through the command line. The online help concerning program usage is

spulse96:Help
USAGE: **spulse96 -d Distance_File [-v] [-t -o -p -i] -a alpha -l L**
[-D -V -A] [-F rfile]
[-m mult] [-STEP|-IMP] [-FUND] [-HIGH] [-Z]
[-EQEX -EXF -ALL] [-LAT] [-2] [-M mode] [-LOCK] [-?] [-h]
Output time series in ASCII file96 format
-d Distance_File Distance control file
This contains one of more lines with following entries:
DIST(km) DT(sec) NPTS T0(sec) VRED(km/s)
The first time point is T0 + DIST/VRED
VRED=0 means infinite velocity though
TIME FUNCTION SPECIFICATION
-t Triangular pulse of base 2 L dt
-p Parabolic Pulse of base 4 L dt
-l L (default 1)duration control parameter
-o Ohnaka pulse with parameter alpha
-i Dirac Delta function
-a alpha Shape parameter for Ohnaka pulse
-F rfile User supplied pulse
-m mult Multiplier (default 1.0)
-Z (default false) zero phase triangular/parabolic pulse
By default the source time function is
-STEP (default) steplike integral of aboce pulses

-IMP impulse like pulse with unit area
steplike. **-IMP** forces impulse like. **-D -IMP** is Green s function
These do not define the shape but rather the
shape of the source pulse. For earthquake
studies use the default steplike
OUTPUT and UNITS
-D Output is ground displacment
-V Output is ground velocity (default)
-A Output is ground acceleration
If the model is km, km/s, gm/cm^3 then the output is
Option units
-A cm/s/s for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne
-V cm/s for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne
-D cm for a moment of 1.0e+20 dyne-cm
or a force of 1.0e+15 dyne
In a fluid the stress is in Pa for a
moment of 1.0e+16 dyne-cm or force 1.0e+14 dyne
If the model is MKS, e.g, m, m/s, kg/m^3, then
ZRT Greens functions are for moment 1.0 N-m, force
1.0 N with units m, m/s, m/s/s. The P stress are in Pa
OUTPUT GREENs FUNCTIONS
-EXF Explosion and point force green s functions
-EQEX Earthquake and double couple green s functions
-ALL Earthquake, Explosion and Point Force
-LAT (default false) Laterally varying eigenfunctions
-2 (default false) Use double length internally
OUTPUT MODES
-FUND (default all) fundamental modes only
-HIGH (default all) all higher modes only
-M [nmode] (default all) mode number to compute[0=fund,1=1st]
-LOCK (default false) locked mode used
OTHER
-v Verbose output
-? Write this help message
-h Write this help message

7. sdpsrf96

This programs reads the phase velocity dispersion file created by **sdisp96** and plots the dispersion curves. A listing is optional. The output is a CALPLOT file with name *SDISPL.PLT* or *SDISPR.PLT* for Love and Rayleigh wave dispersion, respectively. If the **-TXT** flag is used, then a listing of the dispersion information is on file *SDISPL.TXT* or *SDISPR.TXT* for Love and Rayleigh waves, respectively. Figure 6 shows the processing

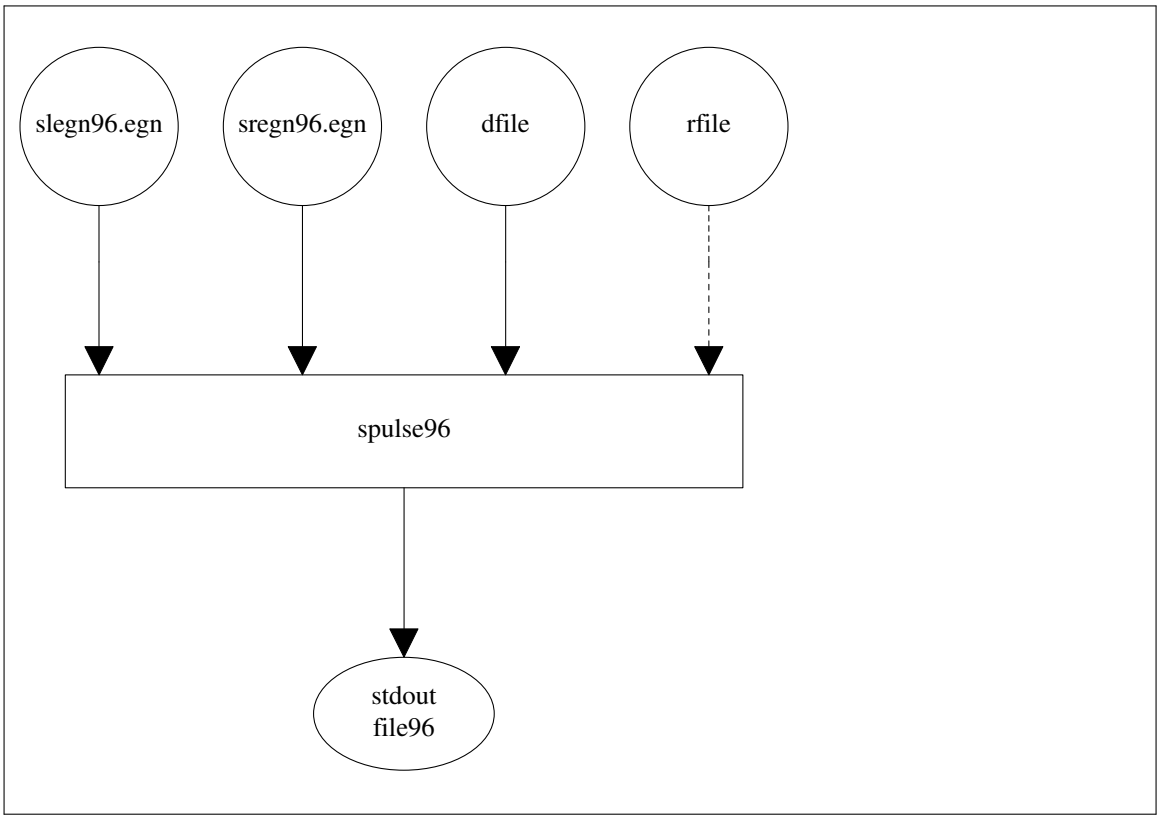


Fig. 5. Processing flow for **spulse96**

flow for this program.

Program control is through the command line:

sdpsrf96 *[flags]*, where the command line flags are

-L Plot the Love wave dispersion contained in the file *sdisp96.lov*.

-R Plot the Rayleigh wave dispersion contained in the file *sdisp96.ray*. br *Note that either* **-L** *or* **-R** *must be specified*, but not both.

-FREQ the X-axis is frequency (default)

-PER the X-axis is period

-XMIN Minimum value of X-axis

-XMAX Maximum value of X-axis *(default is automatic determination of limits)*

-YMIN Minimum value of phase velocity for vertical axis

-YMAX Maximum value of phase velocity for vertical axis

-X0 x0 (default 2.0 in = 5.08 cm)

-Y0 y0 Absolute coordinates of lower left corner of plot (default 1.0 in = 2.54 cm)

-XLEN xlen Length of X-axis (default 6.0 in = 15.24 cm)

-YLEN ylen (default 6.0) Length of Y-axis

-K kolor CALPLOT color code for dispersion curves. Default=1

-NOBOX Do not plot the coordinate axes, only plot corner tics. This is useful for overlaying several plots.

-TXT (default .false.) create a text file listing of the dispersion with the name *SDISPL.TXT* or *SDISPR.TXT* for Love and Rayleigh waves, respectively.

-ASC (default .false.) create an ASCII text file listing of the dispersion with the name

SDISPL.ASC or *SDISPR.ASC* for Love and Rayleigh waves, respectively. These files have a single single line header followed by numeric column entries as in this small exerpt for a

sdpsrf96 -L -ASC

```
LMODE NFREQ       PERIOD(S)      FREQUENCY(Hz)          C(KM/S)
    0     1    2.000000000       0.5000000000       3.553414331
    0     2    2.000977040       0.4997558594       3.553417631
    0     3    2.001955034       0.4995117188       3.553420936
```

Here the columns are mode (with 0 for fundamental), the unique frequency number, the period, frequency and phase velocity.

-T Use the dispersion files *tsdisp96.lov* or *tdisp96.ray* instead of *sdisp96.lov* or *sdisp96.ray*, respectively.

-XLOG X-axis is logarithmic (default is false)

-?

-h On line command help

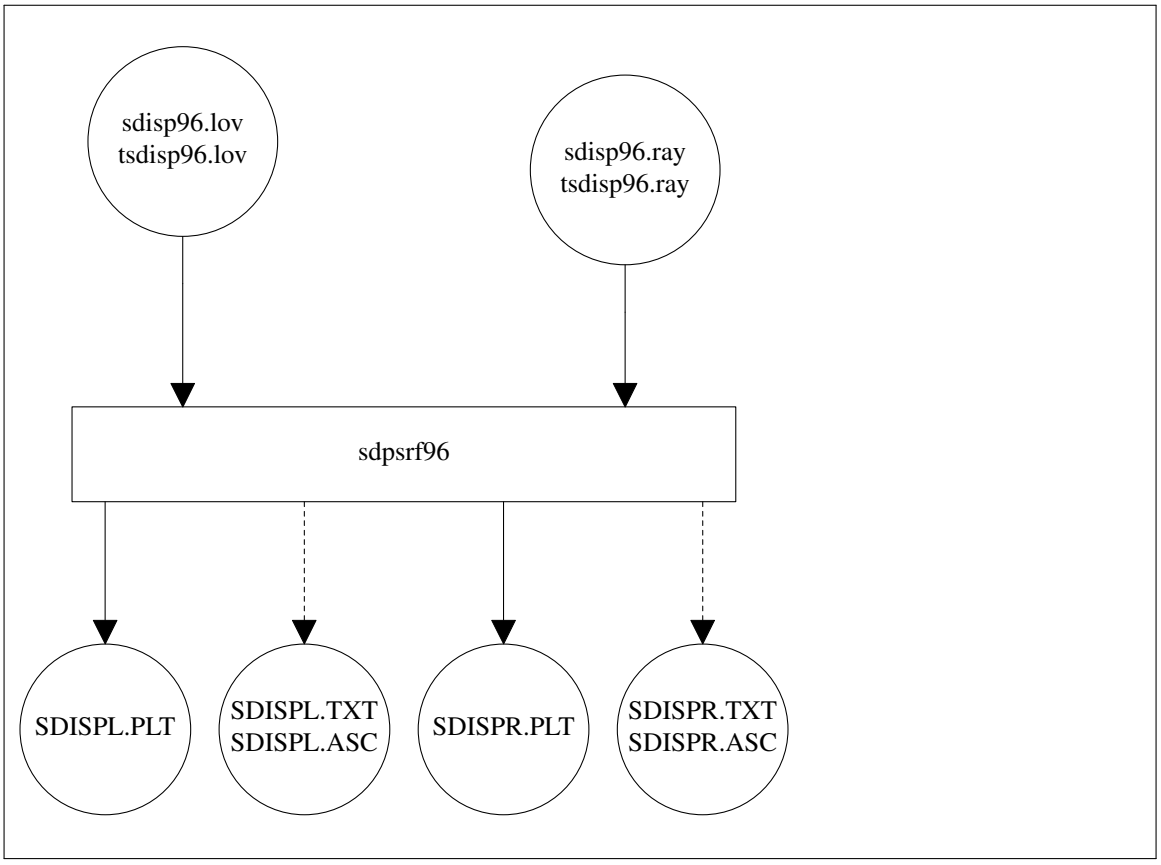


Fig. 6. Processing flow for **sdpsrf96**

8. sdpegn96

This programs reads the eigenfunction file created by **slegn96** or **sregn96** and plots the curves. A listing is also permitted. The output is a CALPLOT file with name *SLEGNU.PLT* or *SREGNU.PLT* for Love and Rayleigh wave group-velocity dispersion, respectively, *SLEGNC.PLT* or *SREGNC.PLT* for Love and Rayleigh wave phase-velocity dispersion, respectively, and *SLEGNG.PLT* or *SREGNG.PLT* for Love and Rayleigh anelastic attenuation coefficient, respectively. Figure 7 shows the processing flow for this program. In the presence of Q, the phase velocities will be those for causal Q, but the group velocities are only for infinite Q.

Program control is through the command line:
sdpegn96 *[flags]*, where the command line flags are

-L Plot the Love wave dispersion contained in the file *sdisp96.lov*.
-R Plot the Rayleigh wave dispersion contained in the file *sdisp96.ray*.
Note that either **-L** *or* **-R** *must be specified*
-U Plot the group velocity dispersion
-C Plot the phase velocity dispersion
-G Plot the anelastic attenuation coefficient
Note that either **-U**, **-C** *or* **-G** must be specified
-A0 AR or AL amplitude factor
-Ac AR sqrt(c) or AL sqrt(c) amplitude factor **-E** Rayleigh ellipticity (Ur/Uz) -R required
-ZR 1/Rayleigh ellipticity (Uz/Ur) -R required
Note one of -C -U or -G is required
-FREQ the X-axis is frequency (default)
-PER the X-axis is period
-XMIN Maximum value of X-axis
-XMAX Minimum value of X-axis
-YMIN Minimum value of phase velocity for vertical axis
-YMAX Maximum value of phase velocity for vertical axis
-X0 x0 (default 2.0)
-Y0 y0 Absolute coordinates of lower left corner of plot (default 1.0)
-XLEN xlen Length of X-axis (default 6.0)
-YLEN ylen Length of Y-axis (default 6.0)
-K kolor CALPLOT color code for dispersion curves. Default=1
-NOBOX Do not plot the coordinate axes, only plot corner tics. This is useful for overlaying several plots.
-TXT (default .false.) create a text file listing of the dispersion in files names *SDISPL.TXT* or *SDISPR.TXT* for Love and Rayleigh waves respectively.
-ASC (default .false.) create an ASCII text file listing of the dispersion in files names *SDISPL.ASC* or *SDISPR.ASC* for Love and Rayleigh waves respectively. These files have a single single line header followed by numeric column entries as in this small exerpt for a

sdpegn96 -R -ASC

RMODE	NFREQ	PERIOD(S)	FREQUENCY(Hz)	C(KM/S)	U(KM/S)	ENERGY	GAMMA(1/KM)	ELLIPTICITY
0	1	2.0000	0.50000	3.2632	3.2640	0.14010E-01	0.48125E-03	0.68118
0	2	2.0010	0.49976	3.2632	3.2640	0.14003E-01	0.48102E-03	0.68118
0	3	2.0020	0.49951	3.2632	3.2640	0.13996E-01	0.48078E-03	0.68118

```
0     4    2.0029      0.49927      3.2632      3.2640      0.13989E-01  0.48055E-03  0.68118
0     5    2.0039      0.49902      3.2632      3.2640      0.13982E-01  0.48031E-03  0.68118
```

Here the columns are mode (with 0 for fundamental), the unique frequency number, the period, frequency, phase velocity, group velocity, energy integral, anelastic attenuation coefficient (**exp(−gammdistance)**, and for Rayleigh waves, the ellipticity..

−XLOG (default linear) X axis is logarithmic

−D *dispfile* (default ignore) plot dispersion values from the file *dispfile* which is in SURF96 dispersion format. Only those values specified by the **−L**, **−R**, **−U**, **−C** and **−G** specific flag combination are plotted. The purpose is to plot observed dispersion on top of model predicted dispersion.

−DT linetype (default solid) linetype= "solid","short" or "long"

−DE *dispfile* (default ignore) plot dispersion values from the file *dispfile* which is in SURF96 dispersion format. Only those values specified by the **−L**, **−R**, **−U**, **−C** and **−G** specific flag combination are plotted. The purpose is to plot observed dispersion and error bars on top of model predicted dispersion.

−S (default no) create output theoretical dispersion values in files in SURF96 format. All values, group velocity, phase velocity and gamma wil be placed into a file named *SLEGN.dsp* or *SREGN.dsp*. Artificial low values will be placed in the error field.

−TICONLY (default false) no numbers on y-axis only tics

−ONLYKM (default false) km/s not U(km/s) C(km/s)

−LAT (default false) use output of slat2d96

−?

−h On line command help

9. sdpder96

This programs reads the depth dependent eigenfunction file created by the **−DER**, **−DH**, **−DA**, **−DB** or **−DR** options of **slegn96** or **sregn96** and plots the curves. A listing is also permitted. The output is a CALPLOT file with name *SLDER.PLT* or *SRDER.PLT* and optionally the ASCII files *SLRDER.TXT* and *SRDER.TXT*, for Love and Rayleigh wave functions, respectively. Figure 8 shows the processing flow for this program.

Program control is through the command line:

sdpder96 *[flags]*, where the command line flags are

−L Plot the Love wave depth dependent functions contained in the file *slegn.der*.

−R Plot the Rayleigh wave depth dependent functions contained in the file *sregn.der*.
Note that either **−L** *or* **−R** *must be specified*

−XMAX Maximum value of X-axis

−XMIN Minimum value of X-axis

−YMIN Minimum value of independent variable

−YMAX Maximum value of independent variable

−X0 x0 (default 2.0)

−Y0 y0 Absolute coordinates of lower left corner of plot (default 1.0)

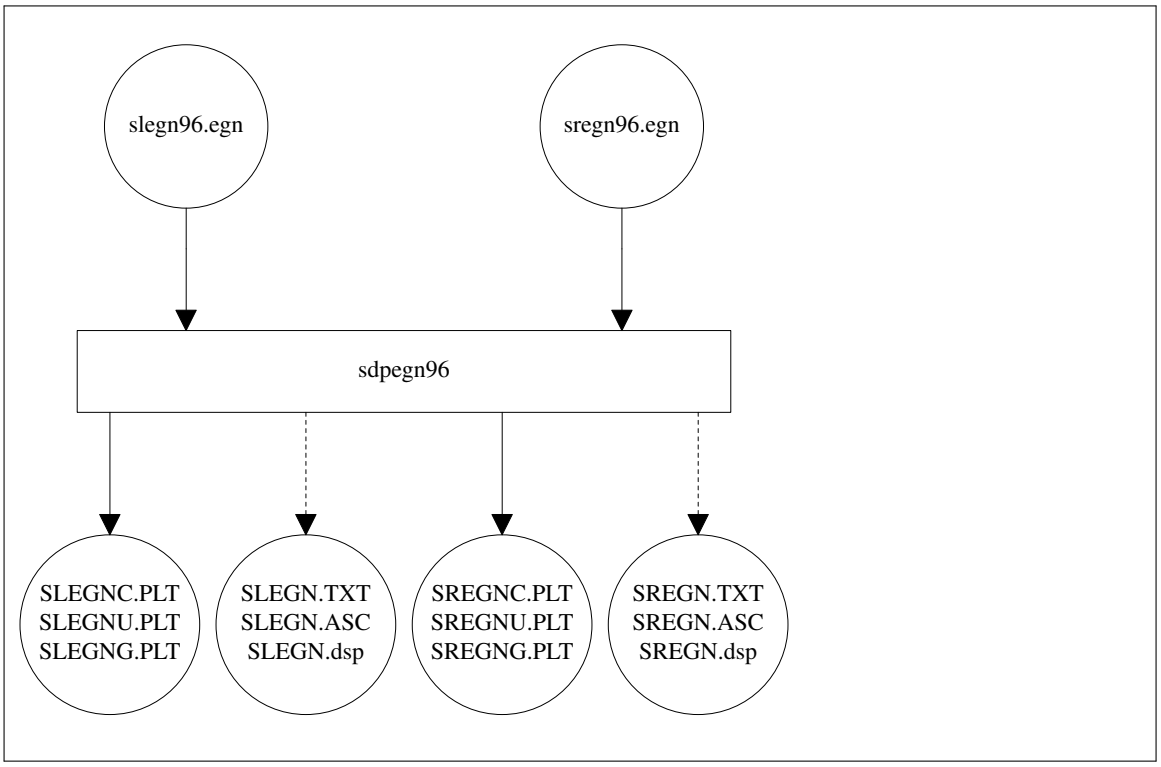


Fig. 7. Processing flow for **sdpegn96**

-XLEN xlen Length of X-axis (default 6.0)
-YLEN ylen Length of Y-axis (default 6.0)
-K kolor CALPLOT color code for dispersion curves. Default=1
-NOBOX Do not plot the coordinate axes, only plot corner tics. This is useful for overlaying several plots.
-CLEAN (default false) No period,mode annotation on the plot
-TXT (default .false.) create a text file listing of the dispersion
-?
-h On line command help

10. sdpdsp96

This programs reads *SURF96* dispersion files and plots them on axes in the same manner as **sdpder96**. The purpose of the program is to compare observations with predictions. The output is a CALPLOT file with name *SLDSPU.PLT* or *SRDSPU.PLT* for Love and Rayleigh wave group-velocity dispersion, respectively, *SLDSPC.PLT* or *SRDSPC.PLT* for Love and Rayleigh wave phase-velocity dispersion, respectively, and *SLDSPG.PLT* or *SRDSPG.PLT* for Love and Rayleigh anelastic attenuation coefficient, respectively. Up to 100 files can be plotted, but this is certainly beyond the capabilities of command line input.

Program control is through the command line:
sdpdsp96 *[flags]*, where the command line flags are

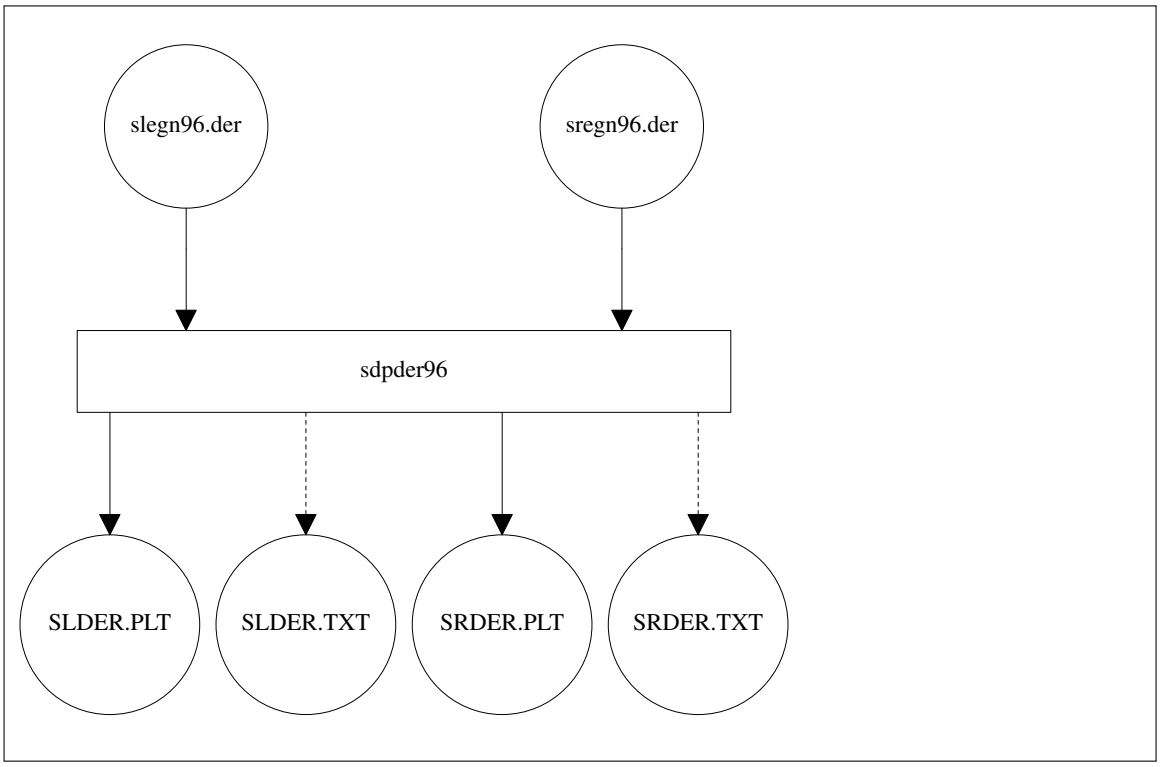


Fig. 8. Processing flow for **sdpder96**

-L Plot the Love wave dispersion contained in the file *sdisp96.lov*.
-R Plot the Rayleigh wave dispersion contained in the file *sdisp96.ray*.
Note that either **-L** *or* **-R** *must be specified*
-U Plot the group velocity dispersion
-C Plot the phase velocity dispersion
-G Plot the anelastic attenuation coefficient
Note that either **-U**, **-C** *or* **-G** must be specified
-FREQ the X-axis is frequency (default)
-PER the X-axis is period
-XMIN Maximum value of X-axis
-XMAX Minimum value of X-axis
-YMIN Minimum value of phase velocity for vertical axis
-YMAX Maximum value of phase velocity for vertical axis
-X0 x0 (default 2.0)
-Y0 y0 Absolute coordinates of lower left corner of plot (default 1.0)
-XLEN xlen Length of X-axis (default 6.0)
-YLEN ylen Length of Y-axis (default 6.0)
-S *symsiz* (default 0.03) size of observed symbol.
-K kolor CALPLOT color code for dispersion curves. Default=1
-NOBOX Do not plot the coordinate axes, only plot corner tics. This is useful for overlaying several plots.
-XLOG (default linear) X axis is logarithmic
-D *dispfile* (default ignore) plot dispersion values from the file *dispfile* which is in SURF96 dispersion format. Only those values specified by the **-L**, **-R**, **-U**, **-C** and

-G specific flag combination is plotted. The purpose is to plot observed dispersion on top of model predicted dispersion. The dispersion values have a size specified by the size parameter below, and a symbol shape keyed on the mode number.

-DE *dispfile* (default ignore) plot dispersion values from the file *dispfile* which is in SURF96 dispersion format. Only those values specified by the **-L**, **-R**, **-U**, **-C** and **-G** specific flag combination is plotted. The purpose is to plot observed dispersion and error bars on top of model predicted dispersion. The dispersion values have a size specified by the size parameter below, and a symbol shape keyed on the mode number.

-DC *dispfile* (default ignore) plot dispersion values from the file *dispfile* which is in SURF96 dispersion format as a *continueous curve. Only those values specified by the* **-L**, **-R**, **-U**, **-C** and **-G** specific flag combination is plotted. Typically this file will be obtained from **sdpegn96**.

-NOBLACK (default black) do not but black around symbol

-V (default false) verbose output

-?

-h On line command help

11. sdprad96

This program plots theoretical radiation pattern plots for a given period, mode and wave type. This program is used as a final step in determining focal mechanisms from surface wave spectral amplitude radiation patterns.

To use this program it is necessary to run the programs **sregn96 -DER** and **slegn96 -DER** to generate the binary files, *sregn96.der* and *slegn96.der*, of eigenfunctions as a function of depth.

The output of the program is a either a *SRADR.PLT* or a *SRADL.PLT* **CALPLOT** file. The plot consists of the radiation pattern with a maximum radius of 1 inch (2.54 cm), the theoretical radiation pattern in black, the observed data points in red, a scale indicating the spectral amplitude in units of *cm-sec*, and the observation period. The command line required a target distance, *dist*. Observed data at a distance, r are first corrected for anelastic attenuation by using the model derived γ by multiplication by the factor $exp(+\gamma r)$ and then propagated to the reference distance my multiplication by the factor $\sqrt{r/dist}$.

If the observed data file is not available, just the theoretical radiation pattern is plotted.

Program control is through the command line:
sdprad96 *[flags]*, where the command line flags are

-DIP dip dip of fault plane

-STK Strike strike of fault plane

-RAKE Rake slip angle on fault plane

-M0 Moment (def=1.0) Seismic moment in units of dyne-cm

-MW *mw* Moment magnitude

-E Explosion

-fx FX -fy Fy -fZ fz
Point force amplitudes (N,E,down) in units of dynes

-XX Mxx -YY Myy -ZZ Mzz -XY Mxy -XZ Mxz -YZ Myz
Moment tensor elements in units of dyne-cm
-DIST dist Normalization distance in km
-HS *hs* Source depth in km
-X0 *x0* (default=1.5) x-coordinate of center of plot
-Y0 *y0* (default=1.75) y-coordinate of center of plot
-O *observed_data* File with observations in MFT96 format
-PER *period* (default 20.0 sec) desired period
-M *mode* (default 0) desired mode (0-Fund)
-L (default Rayl) Plot Love wave radiation
-R (default Rayl) Plot Rayleigh wave radiation
-DMIN *dmin* (default 0 km) minimum for distance sieve
-DMAX *dmax* (default 100000 km) maximum for distance sieve
-A (default false) Plot all periods in one plot
CRACK (default no) use crack model: Dip is dip of
crack, Strike is dip direction of crack rake and Mw or M0, where M0 > 0 and
Rake > 0 for expanding crack
< 0 for closing crack.
M0= sgn(rake) mu DELTA Volume
Poisson ratio = 0.25
-V (default false) Verbose output of theoretical amplitude versus'
-?
-h On line command help

12. Sample Runs

Given the sample model and distance files of Chapter 1, the following commands are run:

```
rbh> sprep96  -HR 0 -HS 10 -M model.d -d dfile  -NMOD 100 -L -R
rbh> sdisp96    > sdisp96.out
rbh> sdpsrf96 -L -XMIN 0 -XMAX 4 -YMIN 3.0 -YMAX 5.0
rbh> sdpsrf96 -R -XMIN 0 -XMAX 4 -YMIN 3.0 -YMAX 5.0
rbh> slegn96    > slegn96.out
rbh> sregn96    > sregn96.out
rbh> spulse96 -d dfile -V -p -l 4  | fprof96
```

The phase velocity dispersion curves computed using **sdisp96** and plotted using **sdpsrf96** are shown in Figure 9 for Love and in Figure 10 for Rayleigh waves.

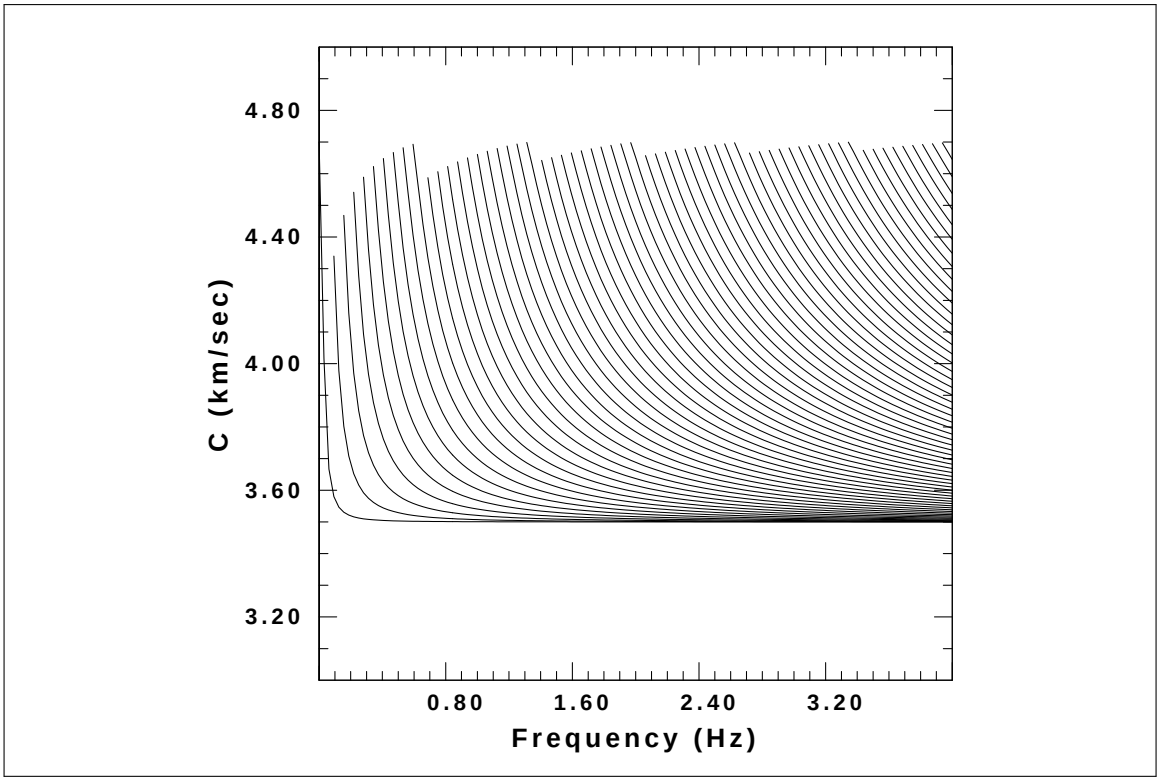


Fig. 9. Love wave phase velocity dispersion curves

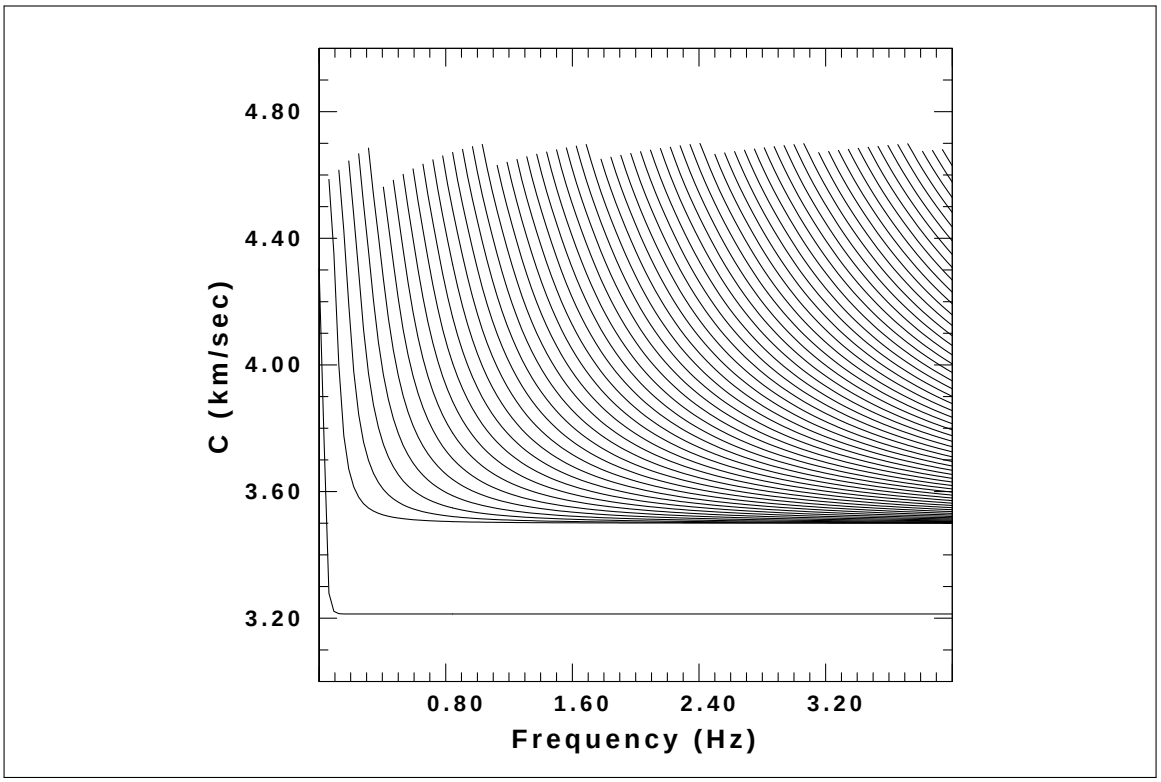


Fig. 10. Rayleigh wave phase velocity dispersion curves

The synthetic transverse time histories for a vertical strike-slip source are given in Figure

11.

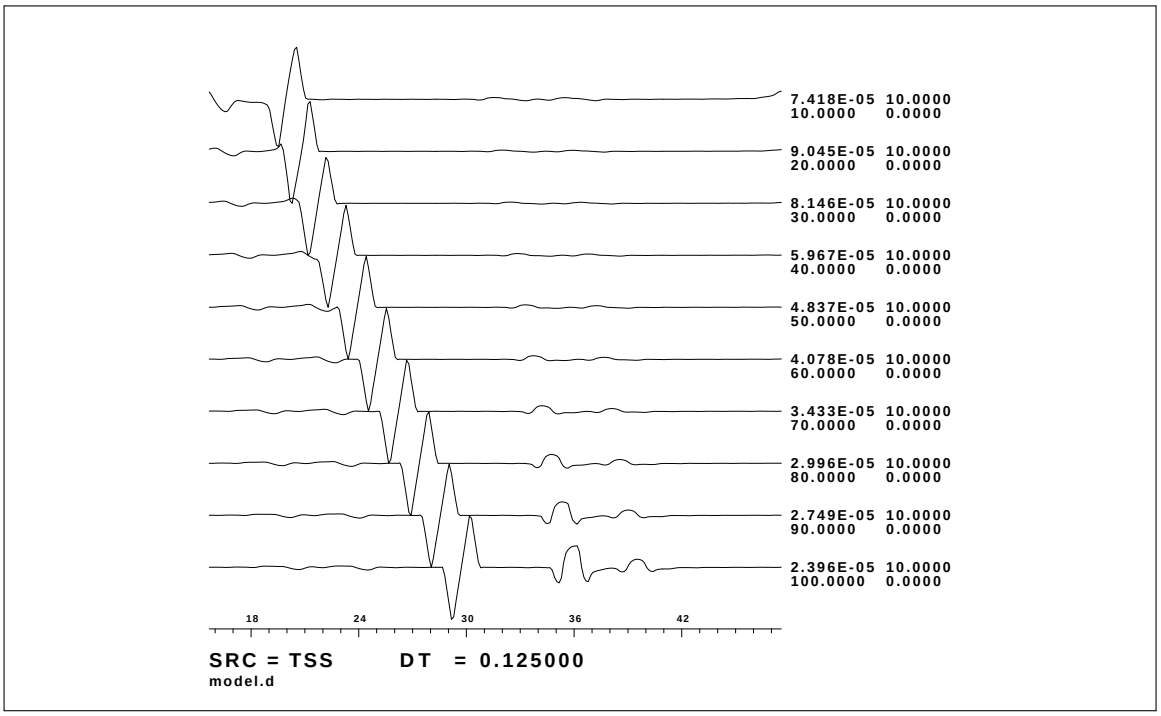


Fig. 11. The TSS time histories file generated by **fprof96**.

The other example shows the use of **sregn96** to generate depth dependent eigenfunctions and partial derivatives for the given model, which are displayed using **sdpder96**. The following test model *model.wat* is used.

```
MODEL
WATER MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
HR       VP       VS   RHO  QP   QS   ETAP  ETAS  FREFP  FREFS
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     1.8      0.0  1.0  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 0.5     6.0      3.5  2.8  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
 1.      8.0      4.7  3.3  0.0  0.0  0.0   0.0   1.0    1.0
```

the following sequence of commands is run:

```
rbh> sprep96 -M nmodel.wat  -DT 5.0 -NPTS 2 -L -R -NMOD 2
rbh> sdisp96 -v
rbh> sregn96 -HS 10 -HR 20 -DER -NOQ
rbh> sdpder96 -R -TXT -K 2 -XLEN 3.0 -X0 1.0 -YLEN 4.0

rbh> reframe -N1 -O < SRDER.PLT > UR.PLT
rbh> reframe -N2 -O < SRDER.PLT > UZ.PLT
rbh> reframe -N3 -O < SRDER.PLT > TZ.PLT
rbh> reframe -N4 -O < SRDER.PLT > TR.PLT
rbh> reframe -N5 -O < SRDER.PLT > DH.PLT
rbh> reframe -N6 -O < SRDER.PLT > DA.PLT
rbh> reframe -N7 -O < SRDER.PLT > DB.PLT
rbh> reframe -N8 -O < SRDER.PLT > DR.PLT

rbh> reframe -O -Y0+5000 -X0+500   < UZ.PLT >  junk
rbh> reframe -O -Y0+5000 -X0+5000  < UR.PLT >> junk
rbh> reframe -O -Y0+5000 -X0+9500  < TZ.PLT >> junk
rbh> reframe -O -Y0+5000 -X0+14000 < TR.PLT >> junk
rbh> reframe -O          -X0+500   < DH.PLT >> junk
rbh> reframe -O          -X0+5000  < DA.PLT >> junk
rbh> reframe -O          -X0+9500  < DB.PLT >> junk
rbh> reframe -O          -X0+14000 < DR.PLT >> junk
rbh> mv junk ALL.PLT
```

In this sequence, the *-DER* flag to **sregn96** indicates that the depth dependent quantities should be output at each of the initial layer boundaries. The sequence of **reframe** commands are used to extract and reposition each of the depth plots. Figure 12 displays these functions. The *DC/DH* is the partial derivative of the phase velocity with respect to a change in layer thickness.

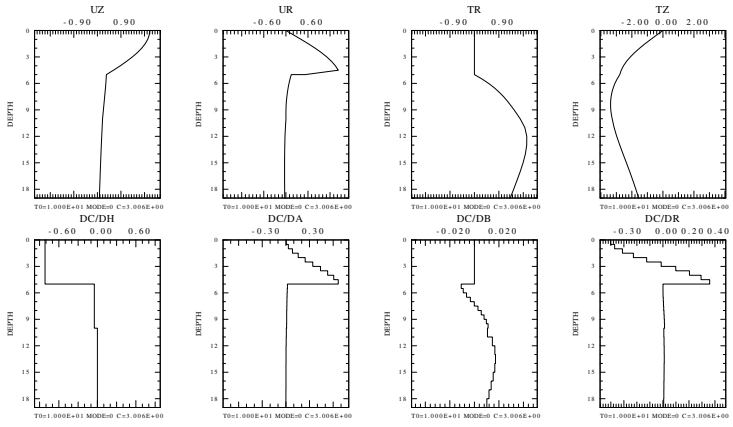


Fig. 12. Display of depth dependent eigenfunctions and phase velocity partial derivatives.

The next example shows how to invoke the program **sdpdsp96** to plot dispersion curves. The files *SREGN.dsp*, *mmsurf96.d* and *msurf96.d* were created by other programs.

The theoretical dispersion *SREGN.dsp* is plotted as a continuous curve (note that the following command is on one line).

```
rbh> sdpdsp96 -R -C -PER -XLOG -YMIN 2.5 -YMAX 5 -DC SREGN.dsp
      -DE mmsurf96.d -DE msurf96.d -XMIN 4 -XMAX 500 -K 1025 -S 0.07
```

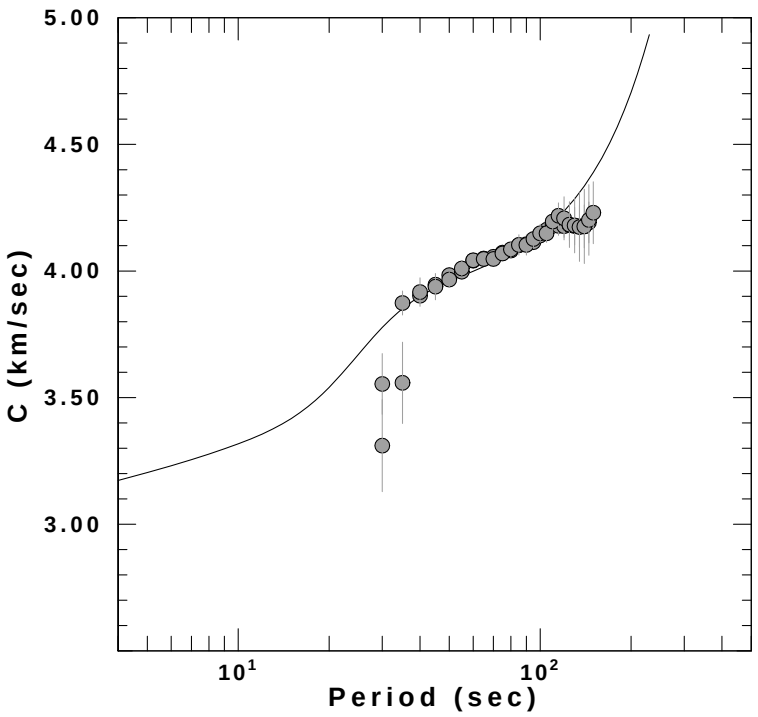


Fig.13. Output of **sdpdsp96**

To plot surface-wave radiation patterns one first uses the program **do_mft** to perform a multiple-filter analysis on surface waves. Assuming that this was done for all stations recording an earthquake, with all the *station.dsp* files concatenated into the single observation file *all.dsp*. The following scripts show how to compute the eigenfunctions as a function of depth and then how to plot the radiation patterns, The resultant plot is shown in Figure 14:

```
#!/bin/sh
#####
#               define the earth model
#####
cat > modcus.d << EOF
MODEL
CUS Model
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
HR      VP      VS   RHO QP   QS   ETAP ETAS FREFP FREFS
1.0000 5.0000 2.8900 2.5000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.1000 3.5200 2.7000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
1.0000 6.4000 3.7000 2.9000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
2.0000 6.7000 3.8700 3.0000 2000 1000 0.0 0.0 1.0 1.0
0.0000 8.1500 4.7000 3.4000 2000 1000 0.0 0.0 1.0 1.0
EOF
######
#               define the desired periods
#####
cat > perfil << EOF
10
20
40
EOF
sprep96 -M modcus.d -HS 10 -HR 0 -L -R -PARR perfil -NMOD 1
sregn96 -DER
slegn96 -DER
```

```
#!/bin/sh
#####
#               define the mechanism, seismic moment and source depth
#####
HS=2
MOM=1.0e+22
DIP=70
RAKE=70
STK=20
#####
#               define reference distance
#####
DIST=1000
MODE=0
#####
#               Loop over Rayleigh Wave Data
#####
for FIG in 01 02 03
do
case ${FIG} in
                01) PER=10 ; X0=2.0 ; Y0=6.0;;
                02) PER=20 ; X0=5.0 ; Y0=6.0;;
                03) PER=40 ; X0=8.0 ; Y0=6.0;;
esac
sdprad96 -R -DIP ${DIP} -RAKE ${RAKE} -STK ${STK} -DIST ${DIST} \
                -PER ${PER} -HS ${HS} -M ${MODE} -M0 ${MOM} -O all.dsp \
                -X0 ${X0} -Y0 ${Y0}
mv SRADR.PLT R${FIG}.PLT
done
for FIG in 04 05 06
do
case ${FIG} in
                04) PER=10 ; X0=2.0 ; Y0=2.0;;
                05) PER=20 ; X0=5.0 ; Y0=2.0;;
                06) PER=40 ; X0=8.0 ; Y0=2.0;;
esac
sdprad96 -L -DIP ${DIP} -RAKE ${RAKE} -STK ${STK} -DIST ${DIST} \
                -PER ${PER} -HS ${HS} -M ${MODE} -M0 ${MOM} -O all.dsp \
                -X0 ${X0} -Y0 ${Y0}
mv SRADL.PLT L${FIG}.PLT
done
#####
#               use calplt to annotate the final figure
#####
calplt  << EOF
NEWPEN
1
CENTER
5.0 3.5 0.20 'Love' 0.0
CENTER
5.0 7.5 0.20 'Rayleigh' 0.0
PEND
EOF
#####
#               concatenate all invividual figures to make
#               final display figure
#####
cat L??.PLT R??.PLT CALPLT.PLT > SRAD.PLT
#####
#               clean up
#####
rm -f CALPLT.PLT CALPLT.cmd L??.PLT R??.PLT
```

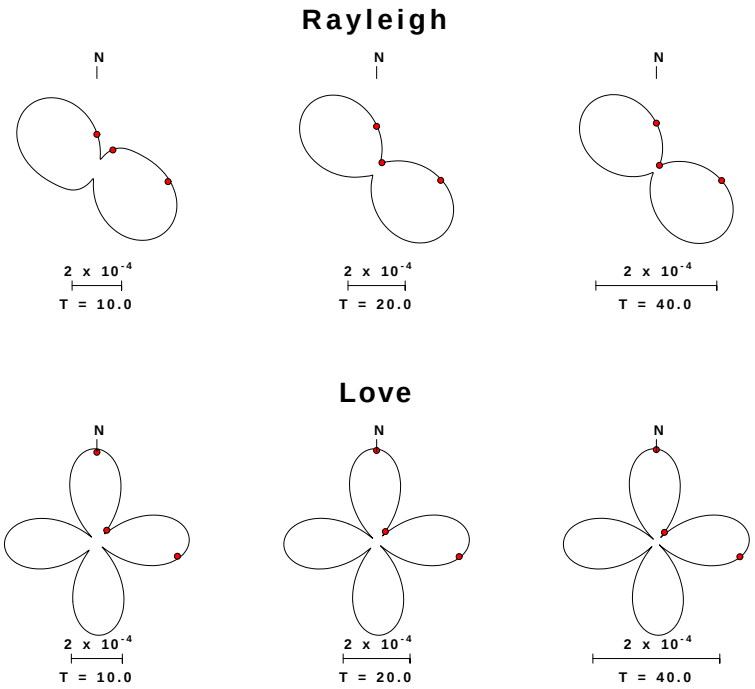


Fig.14. Output of **sdprad96**

13. Dispersion Repair

As mentioned in the discussion of **scomb96**, there may be times that the dispersion following technique used in **sdisp96** fails. This is not necessarily an imperfection in the latter program, but rather reflects a tradeoff between a desire for rapid root determination and accuracy. The following example illustrates the use of **scomb96**.

Low velocity zones in the model severely challenge the dispersion following technique used in *disp96*. In addition, at high frequencies the dispersion curves flatten significantly a phase velocities equal to layer velocities. The following test model *lmodel.d* is used.

```
MODEL
Challenging Model
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
HR      VP      VS   RHO QP   QS   ETAP ETAS FREFP FREFS
10.     6.5     3.8 2.7 0.0 0.0 0.0  0.0  1.0   1.0
10.     6.0     3.5 2.5 0.0 0.0 0.0  0.0  1.0   1.0
10.     7.5     4.4 3.1 0.0 0.0 0.0  0.0  1.0   1.0
10.     7.0     4.1 2.9 0.0 0.0 0.0  0.0  1.0   1.0
00.     8.0     4.7 3.3 0.0 0.0 0.0  0.0  1.0   1.0
```

the following sequence of commands is run:

```
rbh> sprep96 -HS 10 -HR 0 -M lmodel.d -d dfile -L  -R -NMOD 400

rbh> sdisp96 -v

rbh> sdpsrf96 -L -XMIN 0 -XMAX 4 -YMIN 3.0 -YMAX 5.0
plot_nps -F7 -W10 -EPS < SDISPL.PLT > 04011.eps

rbh> sdpsrf96 -R -XMIN 0 -XMAX 4 -YMIN 3.0 -YMAX 5.0
plot_nps -F7 -W10 -EPS < SDISPL.PLT > 04012.eps

rbh> scomb96 -XMIN 3.9 -XMAX 4.0 -CMIN 4.3 -CMAX 4.7 -R

rbh> mv tsdisp96.ray sdisp96.ray

rbh> scomb96 -XMIN 1.75 -XMAX 1.85 -CMIN 4.1 -CMAX 4.2 -R -I

rbh> sdpsrf96 -R -T -XMIN 0 -XMAX 4 -YMIN 3.0 -YMAX 5.0
plot_nps -F7 -W10 -EPS < SDISPL.PLT > 04013.eps
```

The first command creates the data file *sdisp96.dat*. The second command generates the dispersion curves. The third command plots the Love wave dispersion, shown below in Figure 15; the Love wave dispersion has no problems. On the other hand, the plot of the Rayleigh wave dispersion, shown in Figure 16, shows problems. First a mode is missed at high frequency; and secondly, it seems as if two modes are mistakenly identified near 1.8 Hz.

scomb96 is invoked to repair the dispersion near 4.0 Hz. The output is the file *tdisp96.ray*. Since we wish to repair the dispersion curve at another frequency, the UNIX **mv** (or DOS 6.22 **move /Y tsdisp96.ray sdisp96.ray**) renames the program output to be used as input. Since Figure 16 indicated a mistaken duplication of a root near 1.8 Hz, the second use of **scomb96** is told to ignore all previous roots between 4.1 and 4.2 km/sec, and to recompute them. The result of this computation is shown in Figure 17. The dispersion curve has been repaired.

To continue the process and generate synthetic seismograms, the following commands are required:

```
rbh> sregn96 -T
rbh> slegn96 -T
rbh> spulse96 -V -p -l 4 | fprof96
```

Here **sregn96 -T** says to use the *tdisp96.ray* file rather than the incorrect *sdisp96.ray*. Of course one can also rename the file again and just invoke **sregn96**.

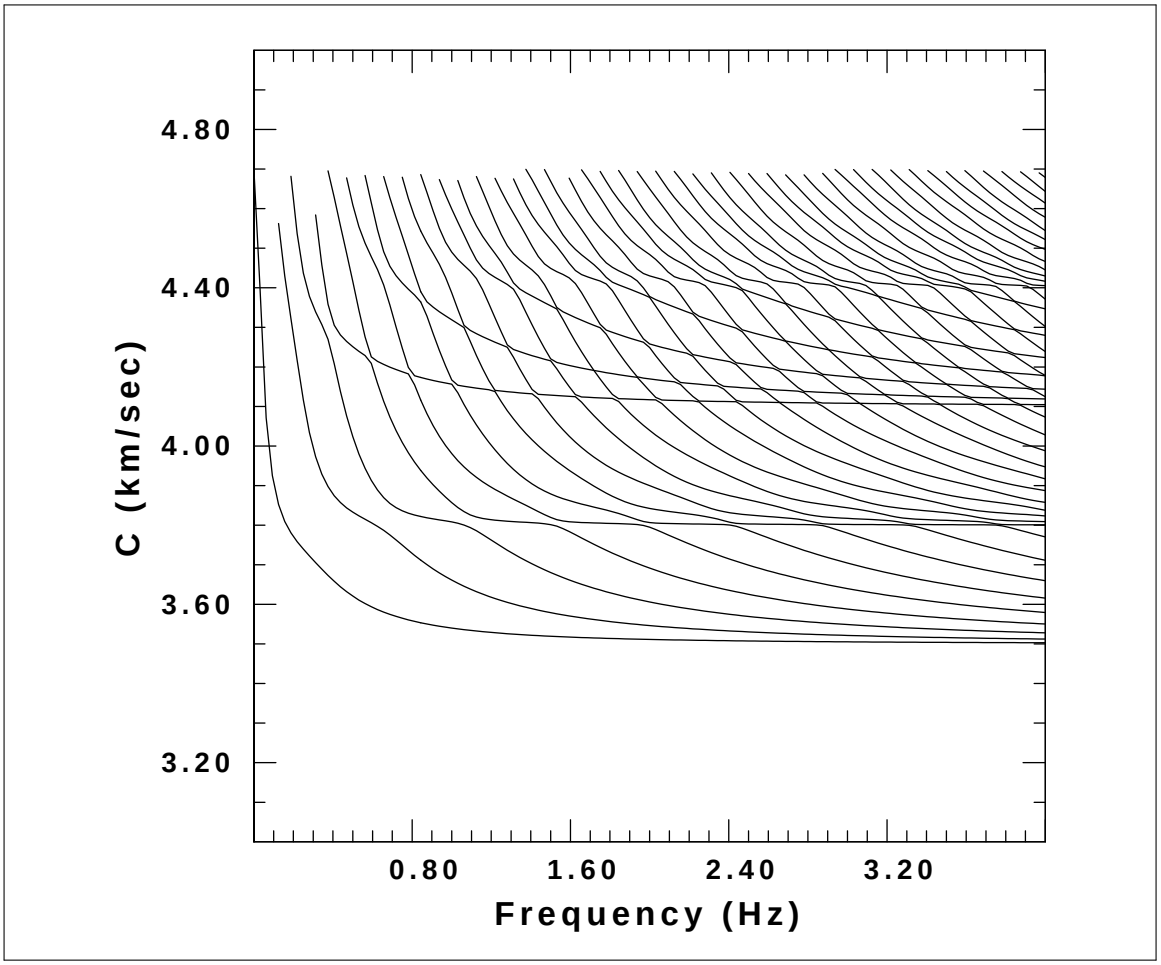


Fig. 15. Love wave phase velocity dispersion curves

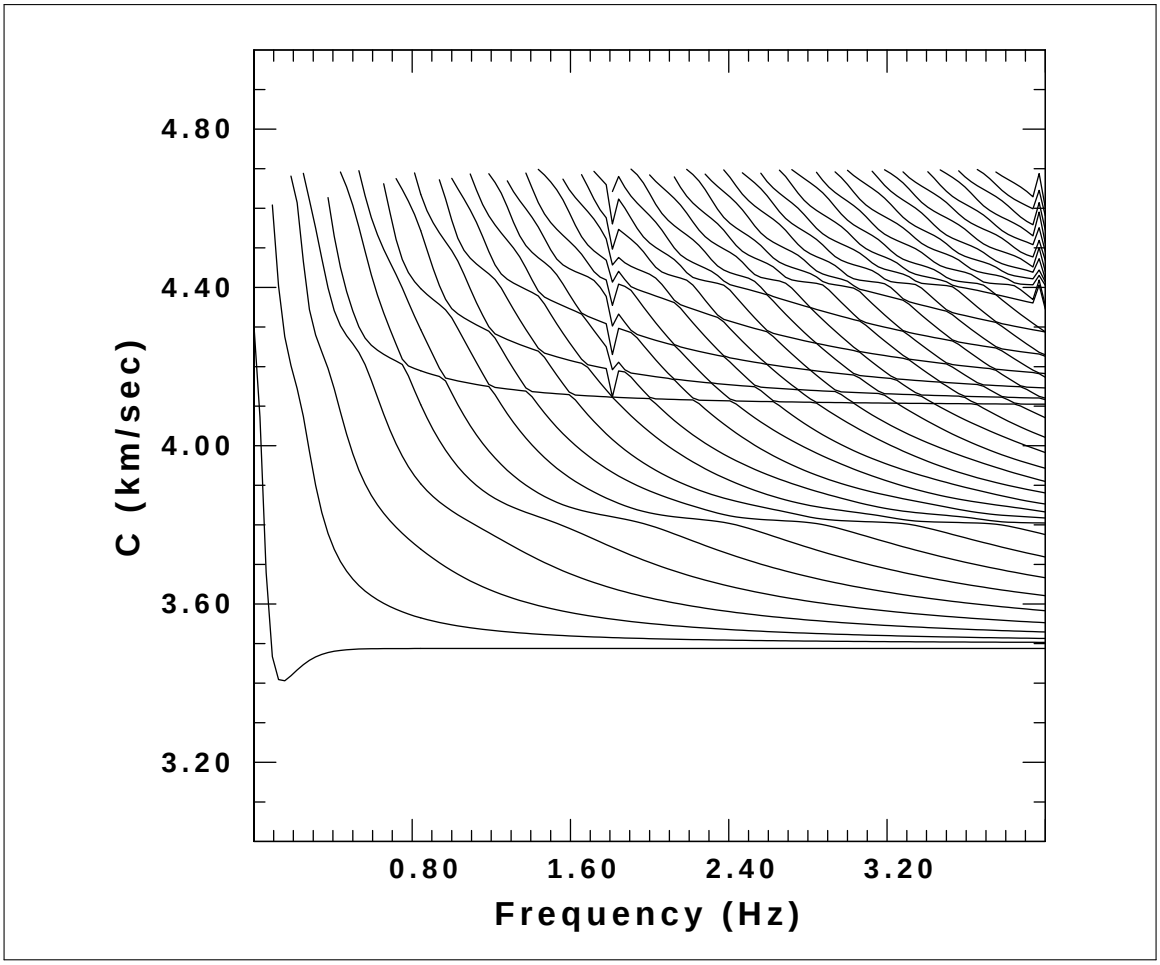


Fig. 16. Rayleigh wave phase velocity dispersion curves showing bad dispersion

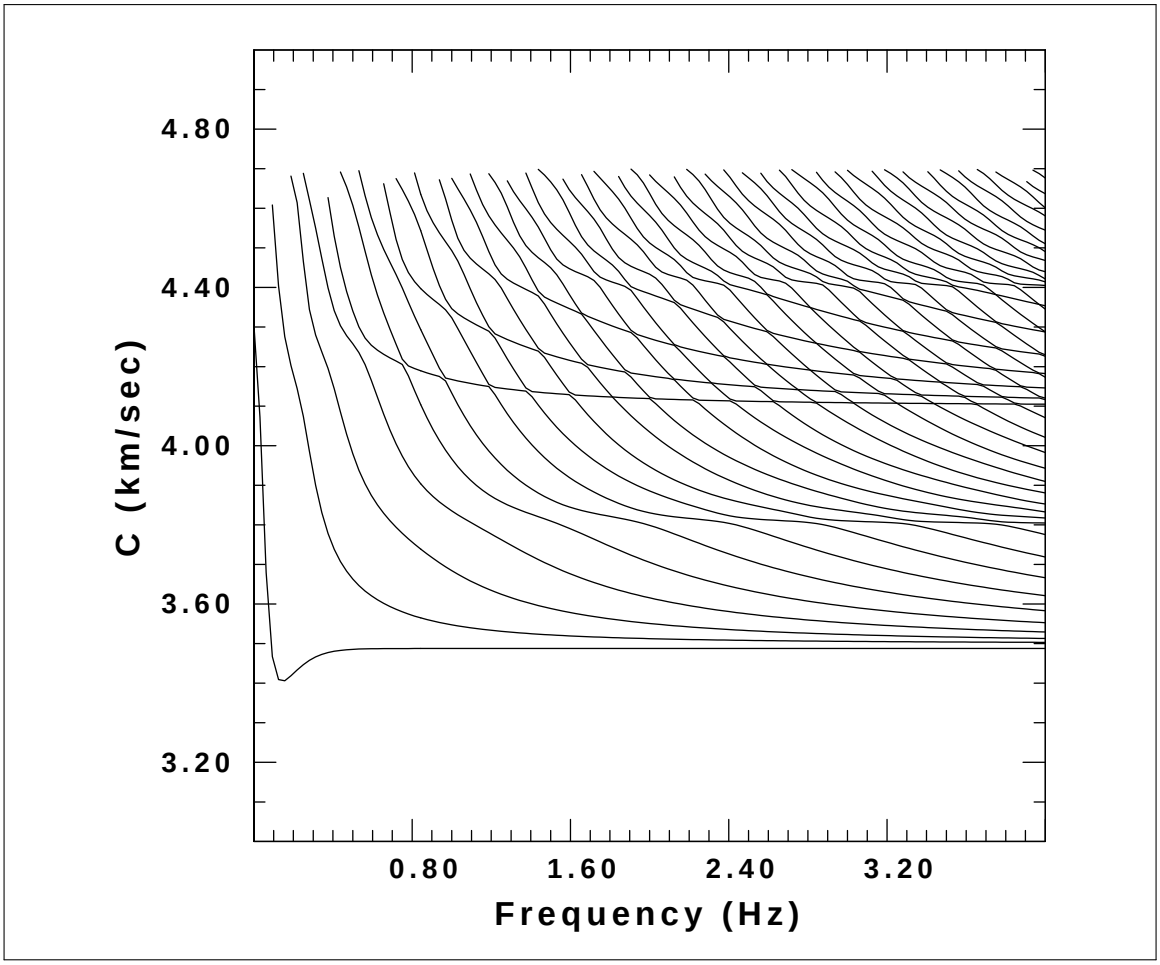
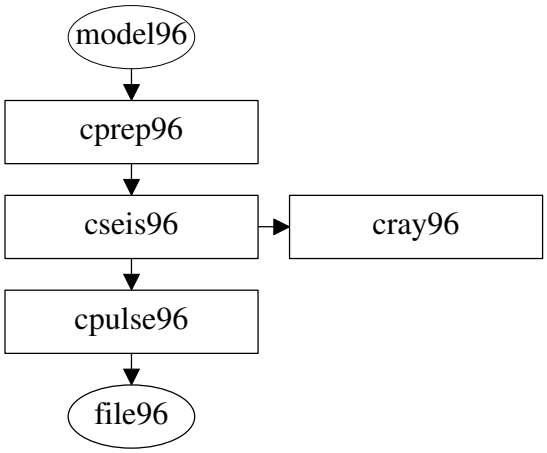


Fig. 17. Repaired Rayleigh wave dispersion curves

CHAPTER 5
ASYMPTOTIC RAY THEORY

1. Introduction

This chapter describes the use of high frequency ray theory to create synthetic seismograms. The program **cseis96** is a slightly modified version of the program **seis81** written by Červený and Pšenčik. There are three stages between the specification of the model file and the final synthetic seismograms. **cprep96** creates a data file *cseis96.dat* for use by **cseis96** which performs the ray tracing and amplitude computations. **cpulse96** uses this information to create the **file96** Green's functions for a specific source time function. The program **cray96** plots the ray trajectories in the medium.



2. cprep96

This program is very similar internally to the program **cprep96** in that it reads an earth model, automatically generates rays, and creates the input data file for the subsequent program. It differs in that the earth model consists of a laterally varying layers and medium parameters.

Figure 1 shows the processing flow for this program. The program requires one earth model file in the **model96** format and two optional files. The output of the program consists of the file *cseis96.dat*, two **CALPLOT** graphics files *CPREP96M.PLT* and *CPREP96R.PLT*, and screen output designated by *stdout*.

Program control is through the command line:

`cprep96` *[flags]*, where the command flags are

`-M model` - *model* is a file in the **model96** format. This is required.

`-DOP`

`-DOSV`

-DOSH

-DOALL

One or more of these must be specified. These tell the program to compute P, SV, SH and all ray segments. If for example, no SH is desired but both P and SV are desired, then use **-DOP -DOSV**. If only **-DOP** is specified, then there will be no P -> SV or SV -> P conversions.

-DOREFL

Permit P-SV conversions only upon reflection

-DOTRAN

Permit P-SV conversions only upon transmission.

-DOCONV

Permit P-SV conversions on both reflection and transmission

The default is that no conversions are computed.

-DENY deny_file

A simple listing of interfaces at which only transmission without wavetype conversion is permitted. Reflections are denied at this boundary. The file consists of a single entry per line, giving the specific boundary number of the earth model.

-R reverberation_file

A simple listing of layers and the maximum number of ray segments in a layer. This is useful to focus on the reverberations within specific layers. The default is to permit as many as possible.

-N maximum_number_of_segments

The maximum number of ray segments permitted in the ray description.

-HS source_depth

The depth of the source in the model

-XS source_x_coordinate

The x-coordinate of the source in the model

-d dfile

(default none) The name of the distance file.

The required distance file. The file contains the following ASCII entries per line:

DIST DT NPTS T0 VRED

where **DIST** is the epicentral distance in kilometers, **DT** is the sampling interval for the time series, **NPTS** is the number of points in the time series (a power of 2). **T0** and **VRED** are used to define the time of the first sample point which is **T0 + DIST/VRED** if **VRED** $\neq$ **0** or **T0** if **VRED = 0**.

-?

-h

Online help

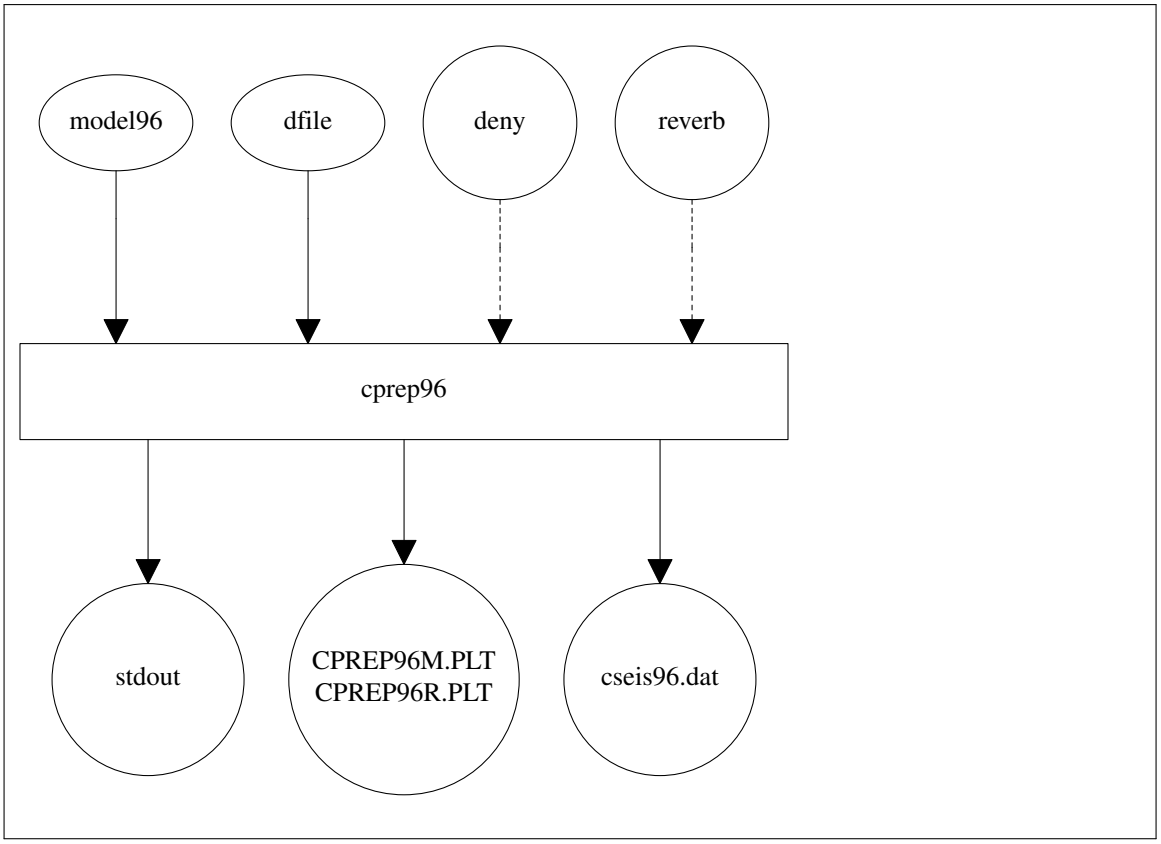


Fig. 1. Processing flow for **cprep96**

3. cseis96

This program performs the dynamic ray tracing through the 2-D medium and computes arrival times and amplitudes for a point source in the 2-D model. This 2.5-D model is equivalent to 3-D wave propagation perpendicular to the axis of symmetry, but essentially in a 2-D medium. *The receiver must be a point at the free surface? There can be no water layers*.

Figure 2 shows the processing flow for this program. Default operation does not require any command line arguments to run, with control from the file *cseis96.dat*.

Program control is through the command line:

`cseis96` *[flags]*, where the command flags are
- **`-v`** (default false) verbose output
- **`-R`** (default false) Generate ray trajectory file to **CRAY96**
- **`-? (default false)`** Online help
- **`-h (default false)`** Online help

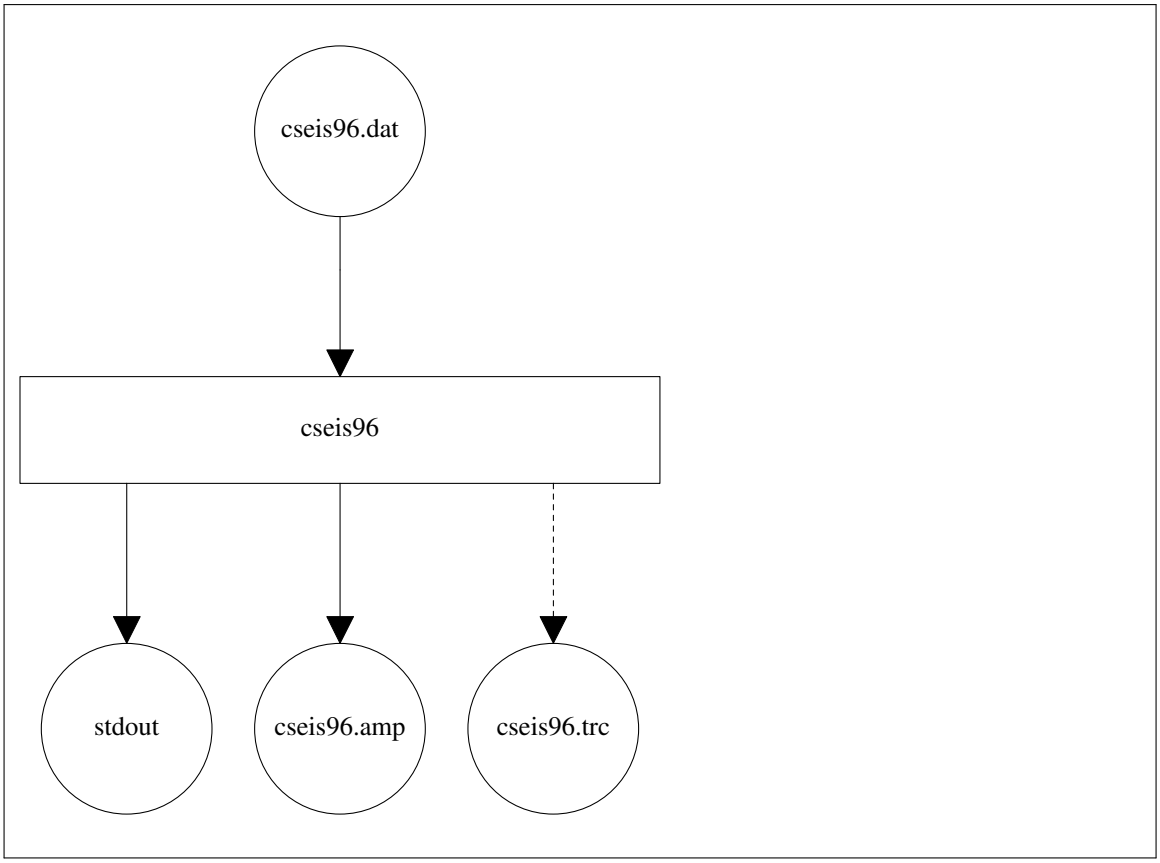


Fig. 2. Processing flow for **cseis96**

4. cpulse96

Figure 3 shows the processing flow for this program. The program requires the *cseis96.amp* file created by **cseis96(V)** and optionally the source pulse definition file *rfile*. The program output is on *stdout* and is a time series in **file96(V)** format.

Program control is through the command line:

`cpulse96` *[flags]*, where the command flags are

`-v`
Verbose output

`-t`
Triangular pulse of base 2 L Δt, which Δt is the sample interval. To avoid problems with sharp truncation in the frequency domain spectra by sampling, never set L < 2. The special case of L = 2 for the triangular pulse is equivalent to the parabolic pulse with L = 1.

`-p`
Parabolic Pulse of base 4 L Δt

`-l L`
Source duration factor for the parabolic and triangular pulses.

-D
Output is ground displacement
-V
Output is ground velocity (default)
-A
Output is ground acceleration
-F rfile
User supplied pulse
-m mult
Multiplier (default 1.0)
-OD
Output is forced to be named displacement
-OV
Output is forced to be named velocity
-OA
Output is forced to be named acceleration
-Q
(default false) do causal Q
-EX (default) Explosion and point force green s functions
-EQ Earthquake and double couple green s functions
-ALL Earthquake, Explosion and Point Force
-Z zero phase triangular/parabolic pulse, else causal
-?
-h
Online help concerning program usage

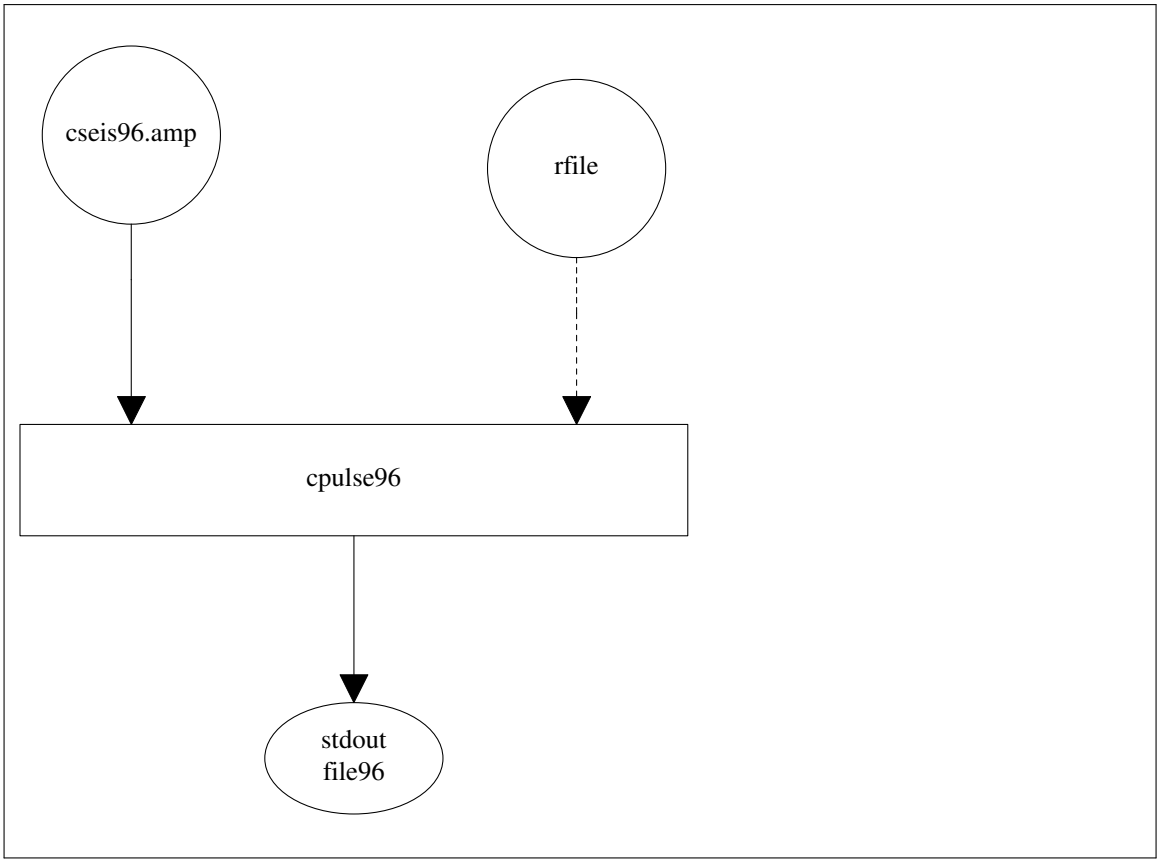


Fig. 3. Processing flow for **cpulse96**

5. cray96

This program plots the ray trajectories found by the program **cseis96**. Figure 4 shows the processing flow for this program. The program requires the *cseis96.trc* file created by **cseis96(V)**. The program output is a **CALPLOT** graphics file *CRAY96.PLT*. The program will operate without any command line arguments, but the user may wish finer control of the plot.

Program control is through the command line:

`cray96` *[flags]*, where the command flags are

`-v`
Verbose output

`-XMIN xmin`
Minimum distance range for plot

`-XMAX xmax`
Maximum depth range for plot

`-ZMIN zmax`
Minimum depth range for plot

`-ZMAX zmax`

Maximum depth range for plot

-?

-h

Online help concerning program usage

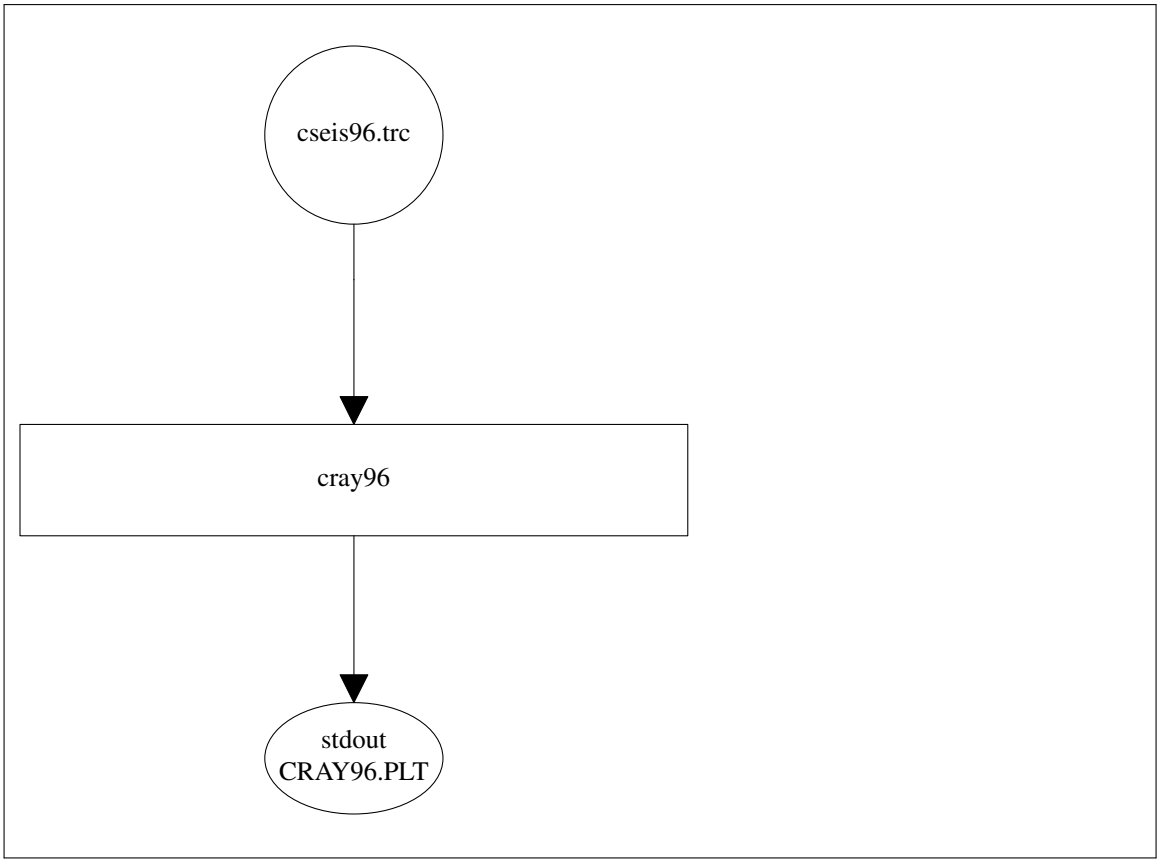


Fig. 4. Processing flow for **cray96**

6. Sample Run

Given the sample model and distance files of Chapter 1, the following commands are run:

```
rbh> cprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10 -DOCONV -d dfile
rbh> cseis96    > cseis96.out
rbh> cpulse96 -V -p -l 4  | fprof96
```

The graphics output of **cprep96** are given in Figures 5 and 6.

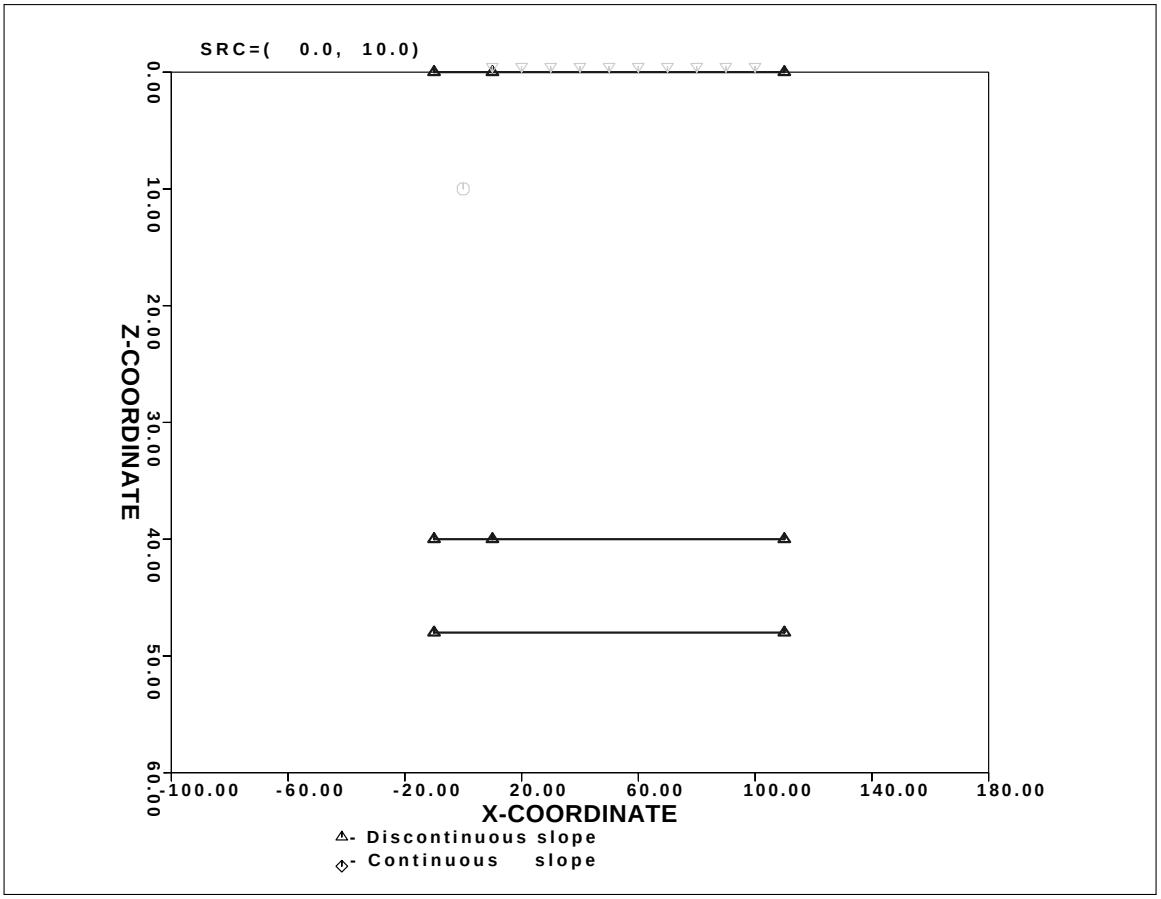


Fig. 5. CPREP96M.PLT which plots the 2-D model.

The synthetic transverse time histories for a vertical strike-slip source are given in Figure 7.

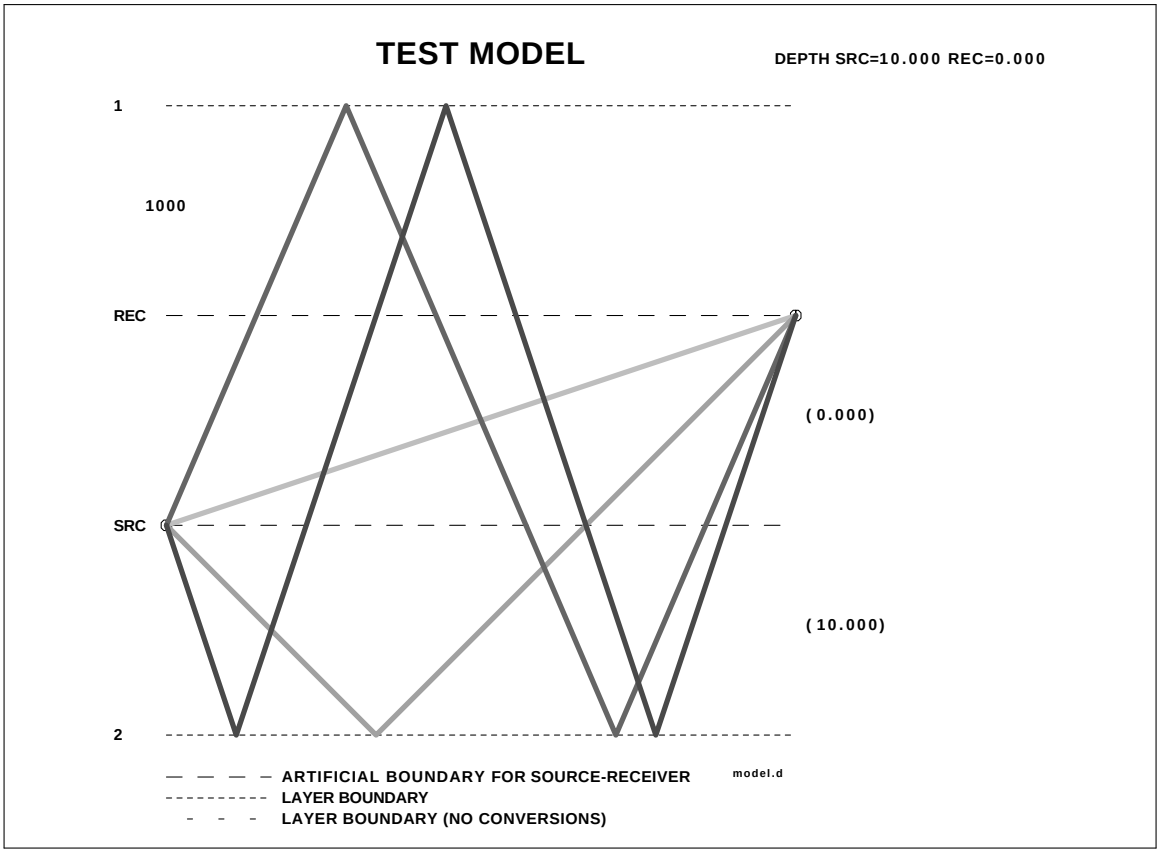


Fig. 6. CPREP96R.PLT which plots the ray description

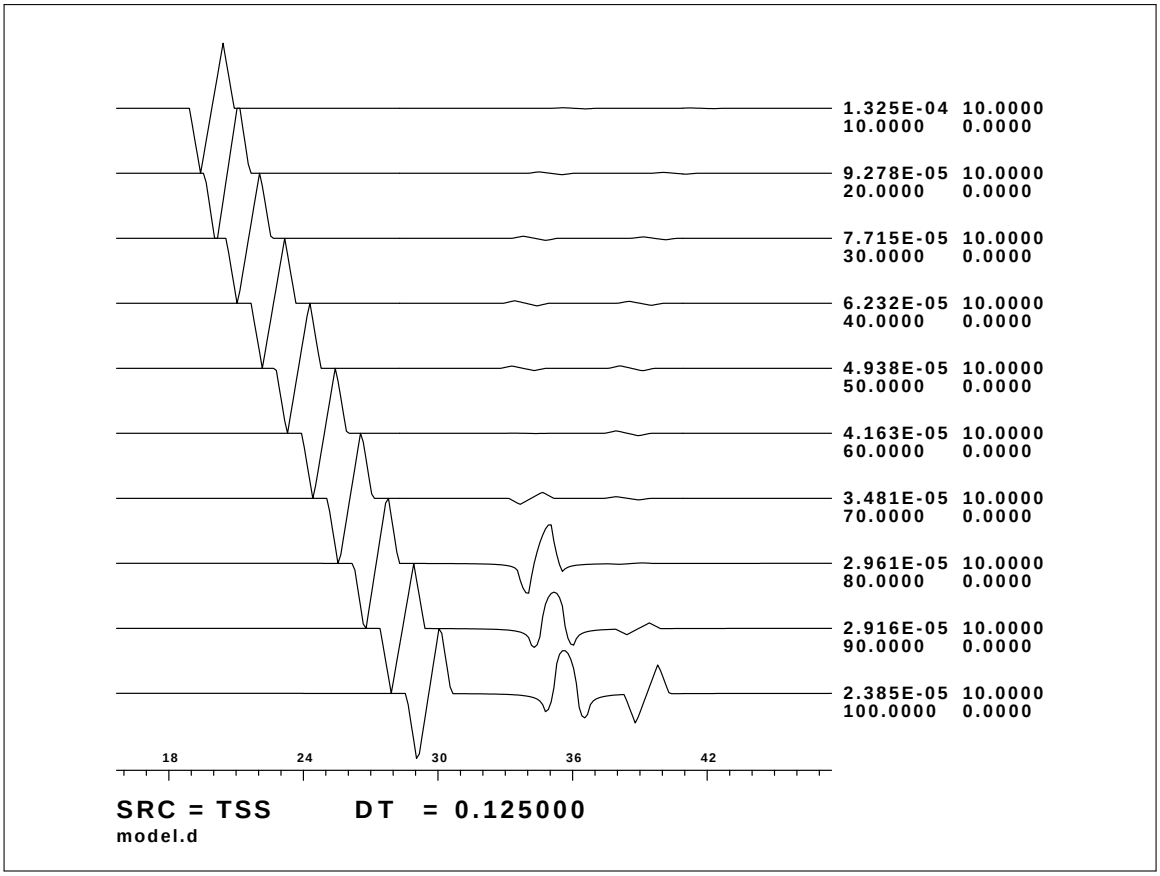


Fig. 7. The TSS time histories file generated by **fprof96**.

To illustrate the output of **cray96**, synthetics are computed only at a distance of 100 km, using the distance file *dfile.100*. Given the sample model and distance files of Chapter 1, the following commands are run:

```
rbh>cprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10 -DOCONV -d dfile.100

rbh>cseis96 -R    > cseis96.out

rbh>cray96
```

The graphics output of **cray96** is given in Figure 8.

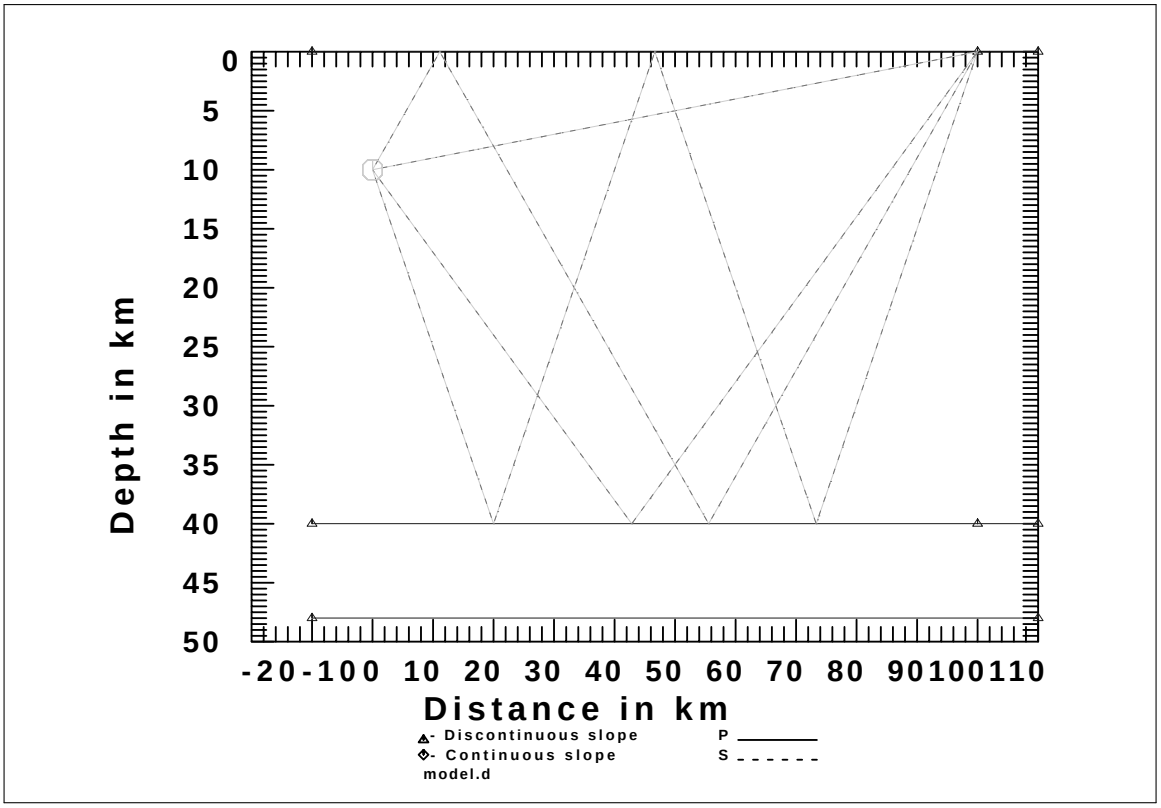


Fig. 8. CRAY96.PLT which plots the 2-D model as well as successful ray paths

7. Laterally Varying Models

The original computer program *seis81* permitted a 2-D laterally varying model and a 2-D velocity function in each layer. The model definition is complicated, especially since the velocity depth function does not have the same value near a boundary since the velocity function is defined on a rectangular grid, upon which the layering is imposed. For simplicity only linear velocity variation is permitted with depth. One can always manually edit the file *cseis96.dat* to create a truly 2-D velocity model by following the documentation of the original program. The following earth models are run using the sequence:

```
rbh>cprep96 -DOALL -HR 0 -HS 10 -M MODEL -N 10 -DOCONV -d dfile.100

rbh>cseis96 -R    > cseis96.out

rbh>cray96
```

Constant Velocity Model - model.1c. This model is

```
MODEL01.1
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
 H       VP       VS   RHO QP   QS   ETAP ETASREFP   REFS
40.      6.0      3.5 2.8 0.0 0.0 0.0  0.0  1.0    1.0
00.      8.0      4.7 3.3 0.0 0.0 0.0  0.0  1.0    1.0
```

The ray traced output is shown in Figure 9.

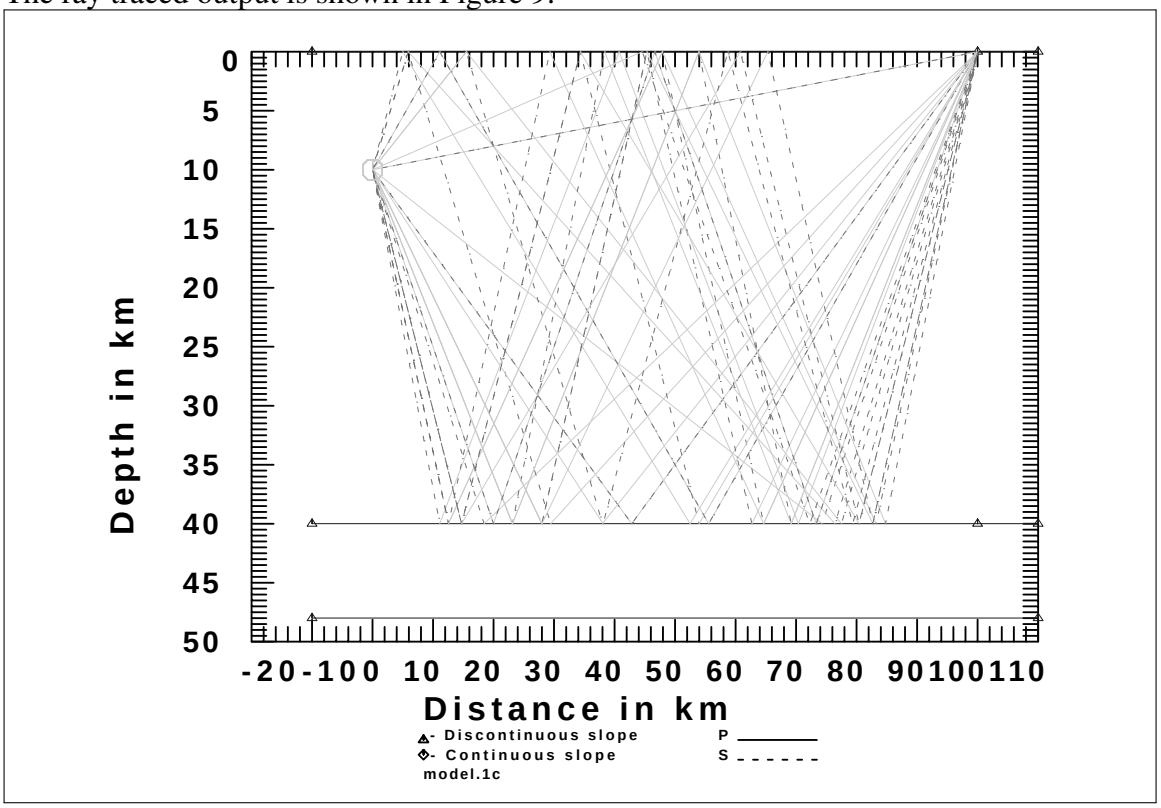


Fig. 9. CRAY96.PLT for a 1-D constant velocity model.

Variable Velocity Model - model.1v. This model is differs from the constant velocity 1-D model by the use of the key word VARIABLE VELOCITY instead of CONSTANT VELOCITY and the fact that each layer is replaced by a dual entry, giving the medium parameters at the top and bottom of each layer.

```
MODEL01.1
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
VARIABLE VELOCITY
LINE08
LINE09
LINE10
LINE11
 H       VP       VS   RHO QP   QS   ETAP ETASREFP   REFS
40.      5.0      3.0 2.2 0.0 0.0 0.0  0.0  1.0    1.0
40.      6.0      3.6 2.5 0.0 0.0 0.0  0.0  1.0    1.0
00.      7.0      4.2 3.8 0.0 0.0 0.0  0.0  1.0    1.0
00.      8.0      4.7 3.3 0.0 0.0 0.0  0.0  1.0    1.0
```

The ray traced output is shown in Figure 10. Note the curved rays.

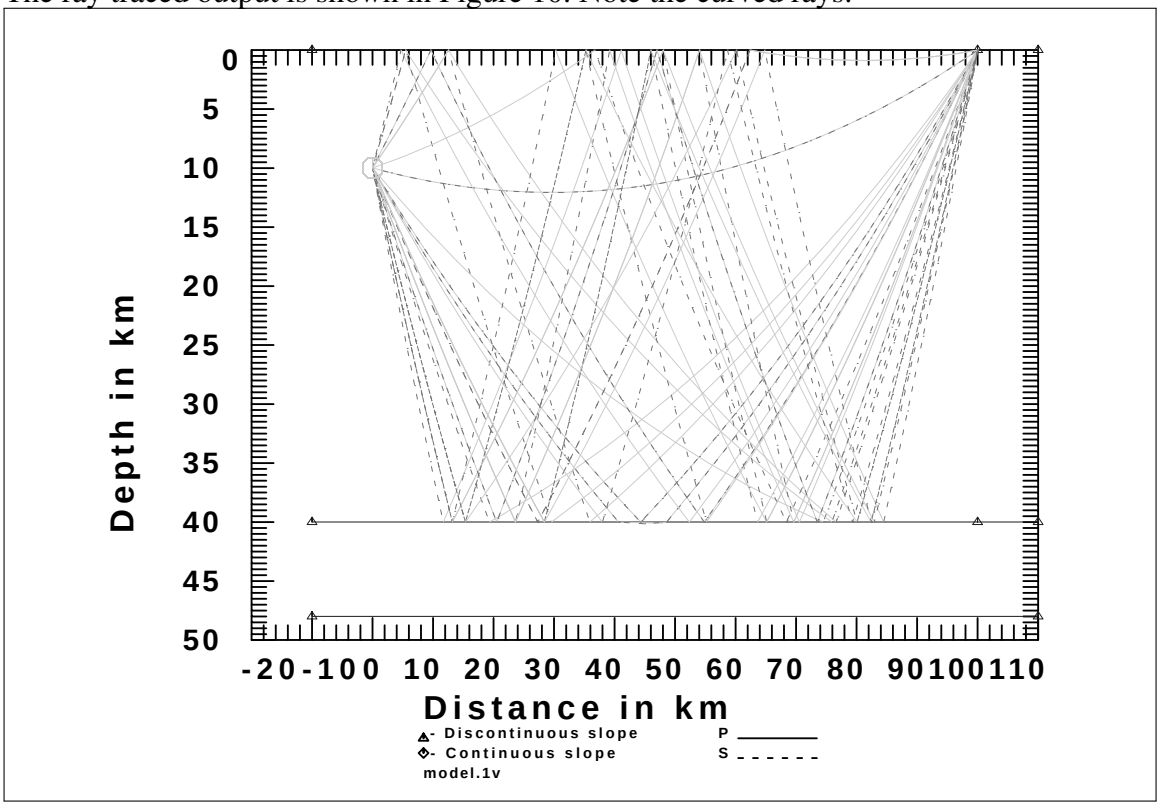


Fig. 10. CRAY96.PLT for a 1-D linear velocity depth model.

Constant Velocity 2-D Model - model.2c. This model is characterized by the key words 2-D and the use of CERVENY for the previously unused LINE11. Following the header is a 2 line comment indicating the format. Next comes the description of the boundaries. The first line gives the number of boundaries (MLYR) and the number of (x,z) pairs defining each boundary (NC(1) .. NC(MLYR)). Then for each layer, NC (x,z) pairs are read in. Following this is the standard 1-D constant velocity model.

```
MODEL01.1
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
2-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
CERVENY
 MLYR (NC(i), i=1,MLYR)
 ((XC(I,J),ZC(I,J),J=1,NC(i)),I=1,MLYR)
 2 2 11
 -10 0 200 0
 -10 40 20 30 40 50 60 40 80 30 100 40 120 50 140 40 160 30 180 40 200 40
 H       VP       VS  RHO QP  QS  ETAP ETASREFP  REFS
40.      5.0      3.0 2.2 0.0 0.0 0.0  0.0  1.0   1.0
00.      8.0      4.7 3.3 0.0 0.0 0.0  0.0  1.0   1.0
```

The ray traced output is shown in Figure 11.

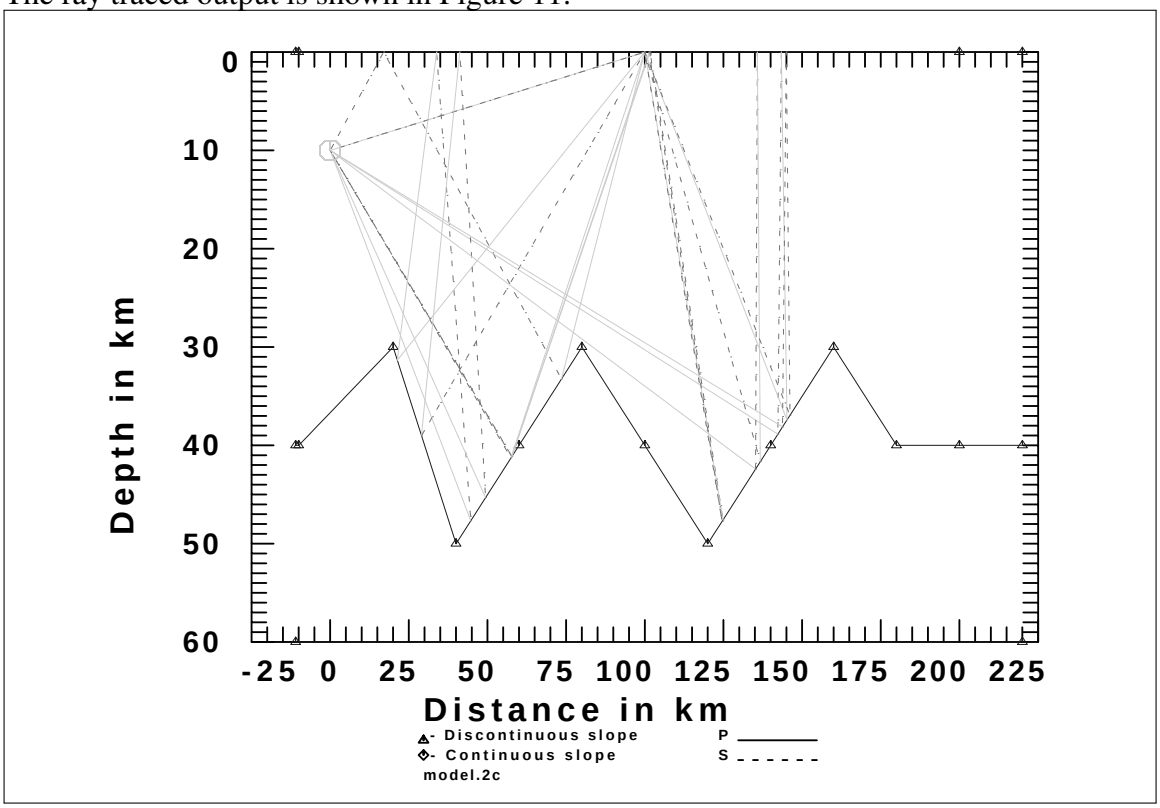


Fig. 11. CRAY96.PLT for a 2-D constant velocity model.

Variable Velocity 2-D Model - model.2v. This model is characterized by the key words 2-D, VARIABLE VELOCITY and the use of CERVENY for the unused LINE11. Following the header is 2 line comment indicating the format. Next comes the description of the boundaries. The first line gives the number of boundaries (MLYR) and the number of (x,z) pairs defining each boundary (NC(1) .. NC(MLYR)). Then for each layer, NC (x,z) pairs are read in. Following this is the variable velocity in Z. indicated by a double entry

for each layer. Because of the way the **seis81** program separates the velocity model and the boundaries, the linear velocity gradient starts at the shallowest depth of the layer above and the deepest depth of the layer beneath.

```
MODEL01.1
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
2-D
VARIABLE VELOCITY
LINE08
LINE09
LINE10
CERVENY
 MLYR (NC(i), i=1,MLYR)
 ((XC(I,J),ZC(I,J),J=1,NC(i)),I=1,MLYR)
 2 2 11
 -10  0 200  0
 -10 40  20 30  40 50  60 40  80 30 100 40
 120 50 140 40 160 30 180 40 200 40
 H       VP       VS  RHO QP  QS  ETAP ETASREFP  REFS
40.      5.0      3.0 2.2 0.0 0.0 0.0  0.0  1.0   1.0
40.      6.0      3.6 2.5 0.0 0.0 0.0  0.0  1.0   1.0
00.      7.0      4.2 3.8 0.0 0.0 0.0  0.0  1.0   1.0
00.      8.0      4.7 3.3 0.0 0.0 0.0  0.0  1.0   1.0
```

The ray traced output is shown in Figure 12.

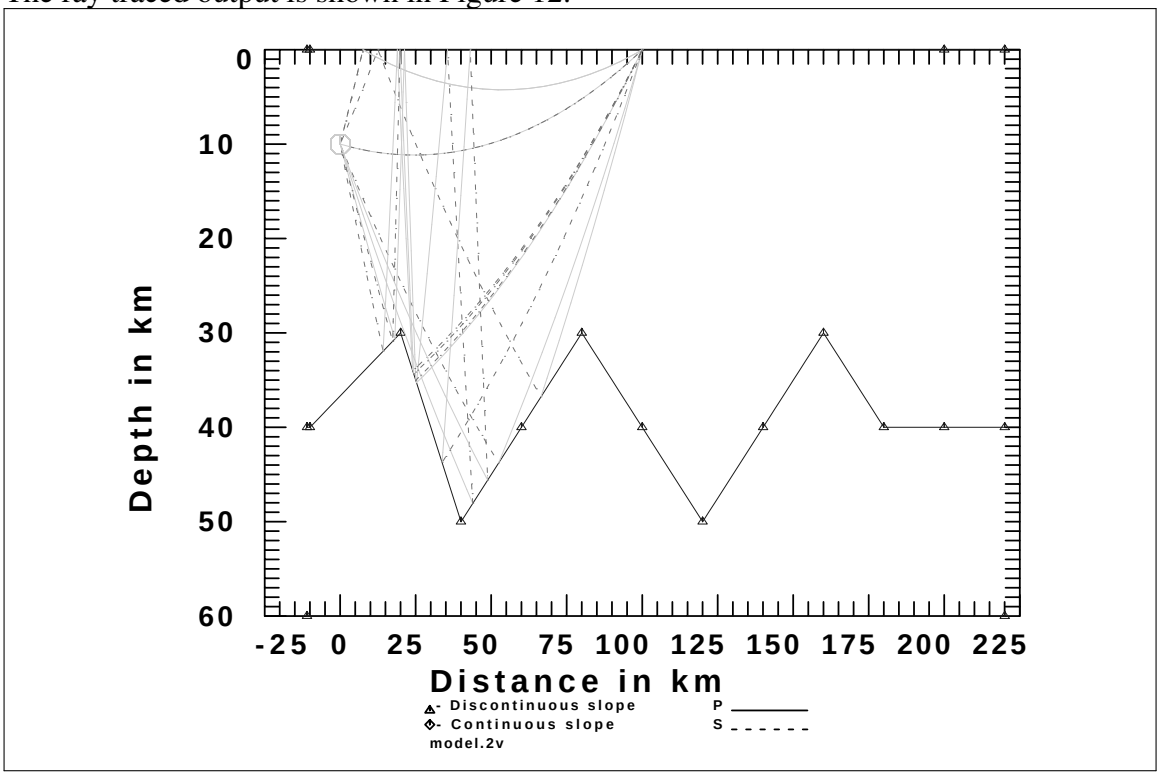


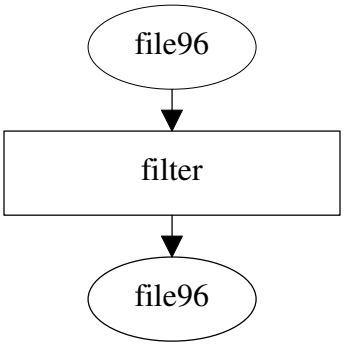
Fig. 12. CRAY96.PLT for a 2-D linear velocity depth model.

CHAPTER 6
FILE96 FILTERS REVISED

1. Introduction

In 1996, when Comptuer Programs in Seismology was developed, the output files were in the *file96* format. Since then **gsac** was written to emulate and extend **sac2000**. Thus many of these operations are more efficiently performed using the **gsac** after converting the *file96* files to SAC files. Thus the program **f96tosac** discussed in the next chapter is most often used.

The first part of this chapter describes the use of routines that act as filters for data in the **file96** trace format. This means that the programs read traces in a **file96** format, modify the traces, and then write the output in a **file96** format. Some of the programs, e.g., **fmech96** change the trace data stream from Green's functions to a three component time history. The unifying feature is that the outputs can be cascaded. thus for example, the sequence of Butterworth filtering and conversion of Green's functions to three component time histories can be interchanged.




```
fbutt96 -fl 0.10 -fh 0.20 -n 3 |
        fmech96 -D 45 -R 90 -S 90 -A 0 -B 180
```

is equivalent to

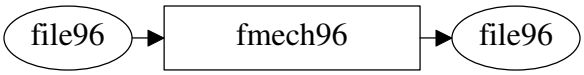
```
fmech96 -D 45 -R 90 -S 90 -A 0 -B 180 |
        butt96 -fl 0.10 -fh 0.20 -n 3
```

although the second will take less computational time since only three traces are filtered instead of up to 16 for each station.

The second part of the chapter, describes plotting routines for the **file96(V)** files. These routines are **fplot96(V)**, **fplot396(V)**, **fplotg96(V)**, and **fprof96(V)**. These routines read a **file96(V)** file and create a **CALPLOT(I)** graphics file.

2. fmech96

This program converts Green's functions into a three component time history for a particular source mechanism - double couple, general moment tensor or point force.



Program control is through the command line:

fmech96 *[flags]*, where the command flags are

-D *dip* dip of fault plane

-S *Strike* strike of fault plane

-R *Rake* slip angle on fault plane

-M0 *Moment (def=1.0)* Seismic moment in units of dyne-cm

-MW *mw* Moment magnitude

-E Explosion

-A *Az* Source to Station Azimuth

-B *Baz* (def=0) Station to Source azimuth

-ROT

Force the three component time histories to be vertical, radial and transverse instead of vertical, north, and east. Since the Green's functions are already vertical, radial, and transverse, the value of the back-azimuth is not used.

-fx FX -fy Fy -fZ fz

Point force amplitudes (N,E,down) in units of dynes

-XX Mxx -YY Myy -ZZ Mzz -XY Mxy -XZ Mxz -YZ Myz

Moment tensor elements in units of dyne-cm

-h

-? Online help

The relation between seismic moment, M_0, and moment magnitude, M_W used is

$$\log_{10} M_0 = 1.5 M_W + 16.05$$

The program will permit a superposition of the point force with any of the moment tensor sources. however, one may wish to do this with separate operations since the source time functions of the point force and moment tensor sources may differ.

The three moment tensor specifications are mutually exclusive: e.g., strike, dip rake; explosion, moment tensor matrix. By this I mean that only one of these specified on the command line will be used to make the three component time history, in fact, the very last entry on the command line controls. Thus the command

```
fmech96 -D 45 -S 120 -R 33 -E
```

will yield the three component time histories for an explosion.

Given the sample model and distance files of Chapter 1, the generalized ray example of Chapter 2, the following commands are executed:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV
rbh> genray96 -d dfile   > genray96.out
rbh> gpulse96 -V -p -l 4 > g.vel
rbh> fmech96 -D 45 -S 45 -R 45 -M0 1.0e+24 -A 0.0 -B 180 < g.vel | fplot96
```

The graphics output of **fmech96** and **fplot96** is given in Figure 1.

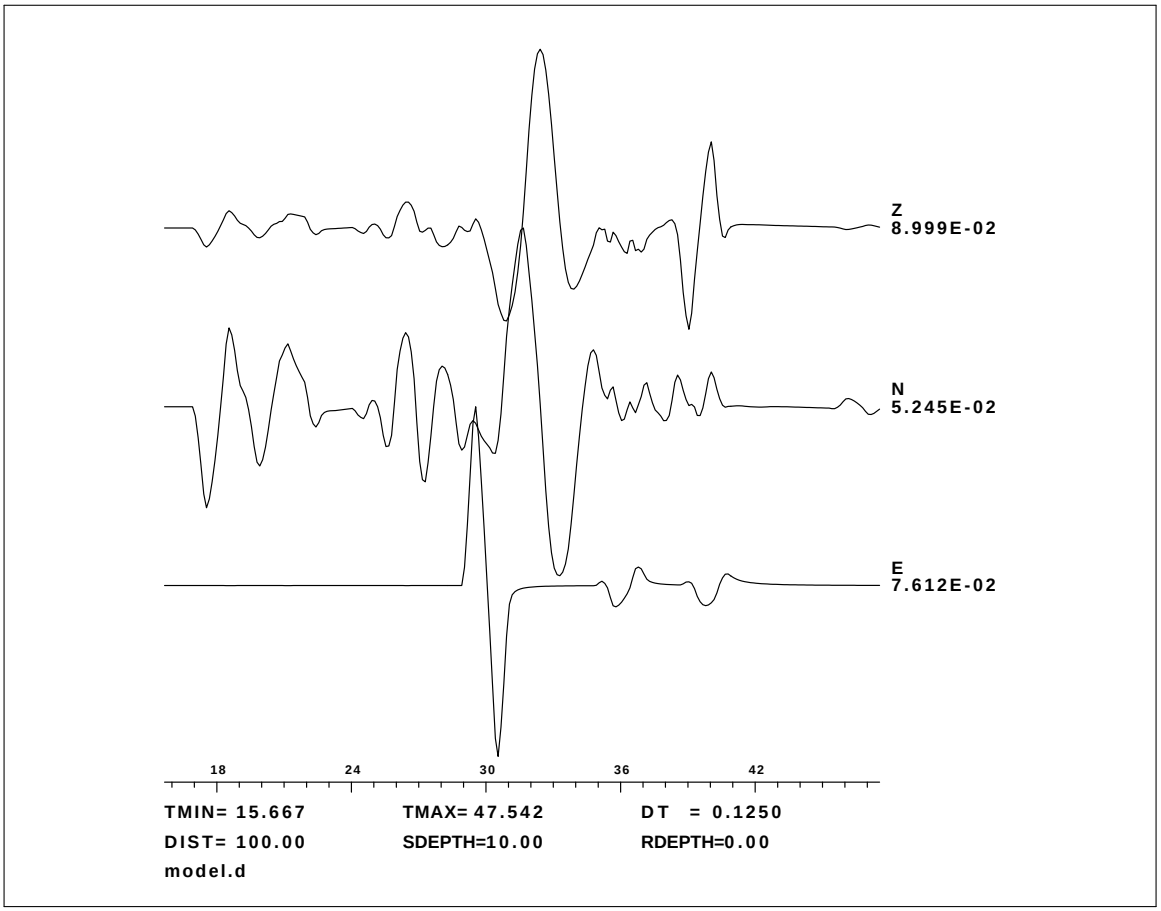
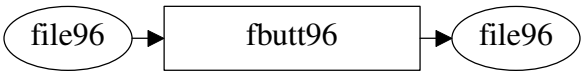


Fig. 1. Result of converting the Green's functions into a three component seismogram. This is the ground velocity in units of *cm/sec* at a distance of 100 km for an source with seismic moment of 1.0×10^{24} dyne-cm for a source pulse with duration of 2 seconds.

3. fbutt96

This program lowpass, highpass or bandpass filters a time history using an n'th order Butterworth filter.



Program control is through command line flags:

fbutt96 *[flags]*, where the command flags are
- **-fl fl** (default 0.0) low cut corner frequency
- **-fh fh** (default Nyquist frequency) high cut corner frequency
- **-n norder** (default 1) order of filter 0 ≤ n ≤10
- **-h**
- **-?** Online help

Given the sample model and distance files of Chapter 1, the the generalized ray run of Chapter 2, following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV

rbh> genray96 -d dfile   > genray96.out

rbh> gpulse96 -V -p -l 4 > g.vel

rbh> fbutt96  -fl 0.10 -fh 0.50 -n 2 < g.vel | fprof96
```

The graphics output of **fbutt96** is given in Figure 2.

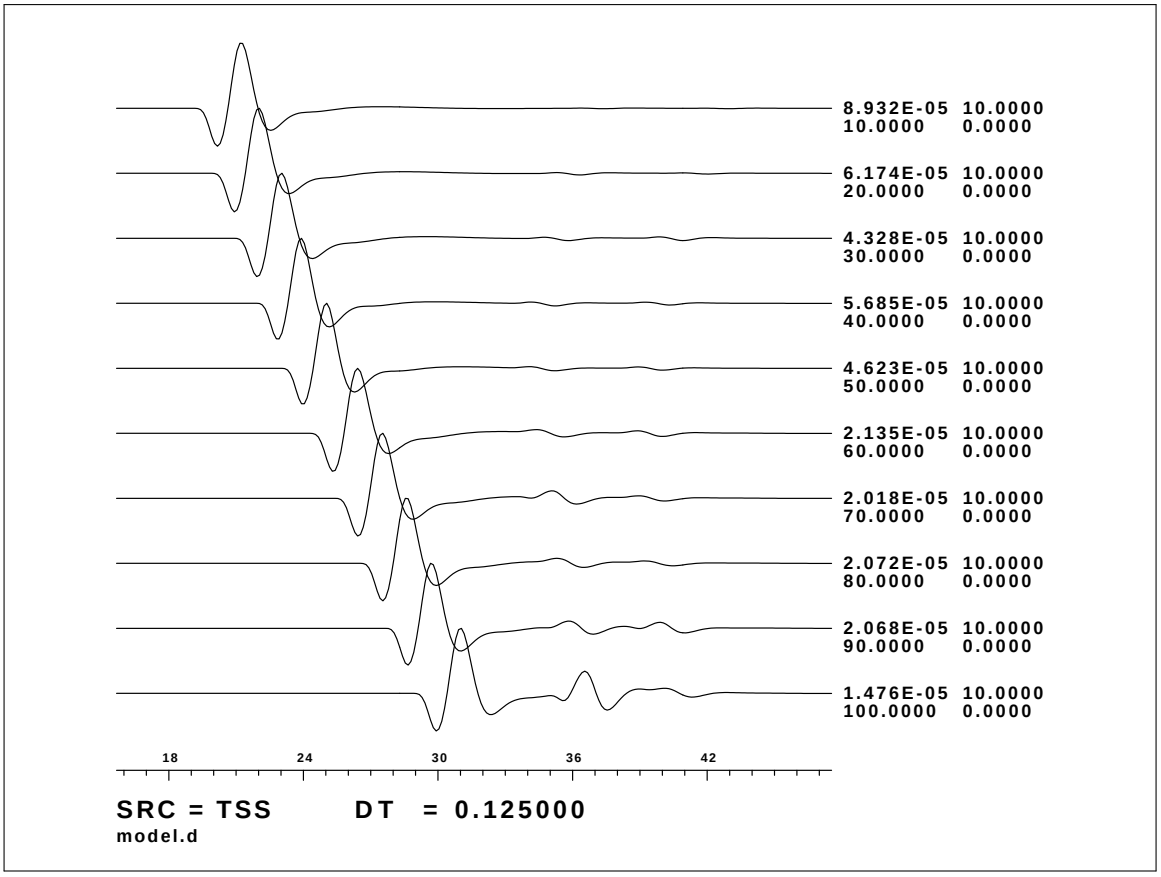
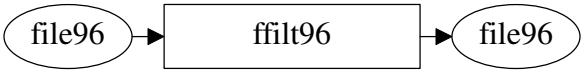


Fig. 2. Result of band pass filtering the ground velocities.

4. ffilt96

This program applies or removes a general instrument/filter response define in terms of poles and zeros. For compatibility with routines that convert SEED or GSE3.0 to SAC, the pole-zero response is defined in SAC format.



Program control is through command line flags:

ffilt96 *[flags]*, where the command flags are
- **-DEMEAN** (default false) remove mean before filter
- **-TAPER** (default false) apply taper before filter
- **-PCT pct** (default 5.0 percent) taper percentage
- **-A** (default true) apply filter
- **-R** (default false) remove filter
- **-PZ pole_zero_file** (none) SAC response file
- **-W water_level** (0.01) to control response removal by avoiding a division by zero

-h
-? Online help concerning program usage

Two examples will be given for this program. The first compares the highpass filter operation of **fbutt96(V)**, performed using a bilinear recursive digital filter on the time series to the results using the discrete Fourier transform technique of **ffilt96(V)**. The SAC pole-zero response file is called *sac.res* and is

```
ZEROS 2
  0.  0.
  0.  0.
POLES   2
  -7.071067  -7.071067
  -7.071067   7.071067
CONSTANT     1.0E+00
```

(Note that the keywords begin in column 1)

Given the sample model and distance files of Chapter 1, the the generalized ray example of Chapter 2, but restricting the synthetic to only one distance at 100 km using the file *dfile.100*, the following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 2  -DOCONV
rbh> genray96 -d dfile.100   > genray96.out
rbh> gpulse96 -V -p -l 4 > file96
```

High pass using recursive filter and pole-zero filter

```
rbh> fbutt96 -n 2 -fl 01.591549 < file96 > file96.b
rbh> ffilt96 -A  -PZ sac.res < file96 > file96.s
```

Compare the time domain recursive Butterworth with the frequency domain filter result in a plot

```
rbh> fplotg96 -f1 file96.b -f2 file96.s
rbh> mv FPLOTG96.PLT FWD.PLT
```

Inverse filter the recursive filter output

```
rbh> ffilt96 -R -PZ sac.res -W 0.001 < file96.b > file96.ss
```

Compare the original timeseries with the inverse filter results

```
rbh> fplotg96 -f1 file96 -f2 file96.ss
rbh> mv FPLOTG96.PLT INV.PLT
```

The comparison of the two high pass filters is given in Figure 3a, while the result of the inverse filtering is given in Figure 3b. The most glaring difference in this latter case is the distortion in the low frequency content of the deconvolved trace since it is impossible to recover the low frequency information lost by the initial high pass filter.

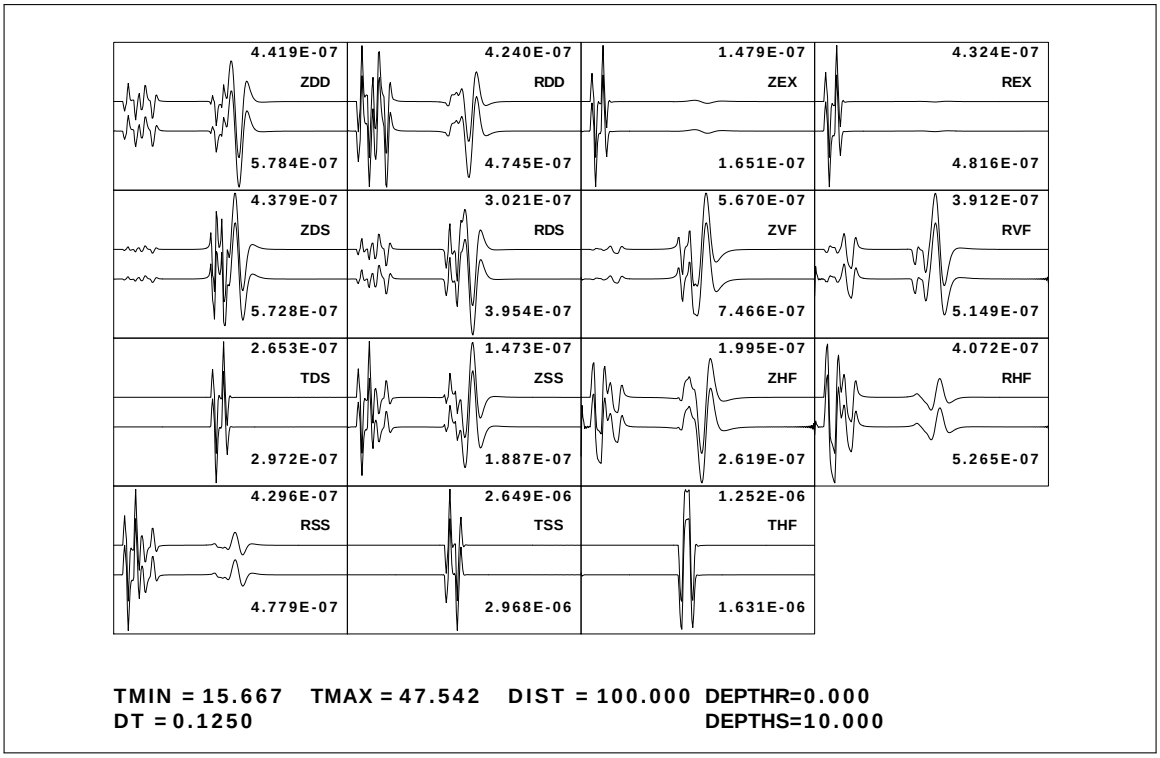


Fig. 3a. Comparison of the time domain filtered (top trace of each Green's function to the pole-zero filtered trace.

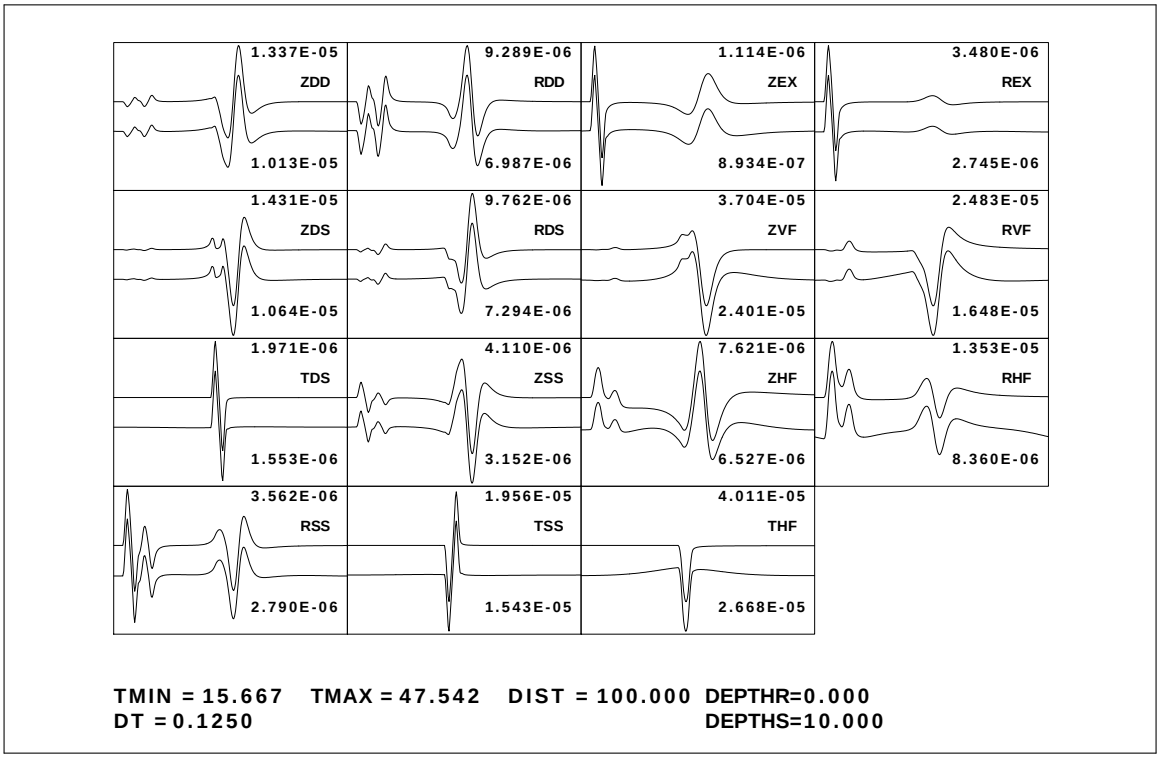


Fig. 3b. Comparison of the initial ground velocity (top trace) to the reconstructed ground velocity (bottom trace).

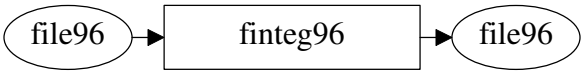
5. finteg96

This program integrates a time history numerically. A simple running sum is used, which is equivalent to a rectangular rule. The algorithm implemented is

$$\int_0^{n\Delta t} x(t)dt = \Delta t \sum_{j=0}^{n} x(j)$$

If there is a DC offset inherent in the synthetic seismogram technique, e.g., modal summation or wavenumber integration which use the discrete Fourier transform, then it may be appropriate to remove a DC trend, defined by the first sample point, prior to summation. The flag **-DC** yields the following approximation to the integral:

$$\int_0^{n\Delta t} x(t)dt = \Delta t \sum_{j=0}^{n} \left[x(j) - x(0) \right]$$



Program control is through the command line:

finteg96 *[flags]*, where the command flags are

-DC - Remove the DC offset prior to performing the integration. *(default - DC offset is not removed)*

-h

-?

Online help

Given the sample model and distance files of Chapter 1, the the generalized ray run of Chapter 2, following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV

rbh> genray96 -d dfile   > genray96.out

rbh> gpulse96 -V -p -l 4 > g.vel

rbh> finteg96   < g.vel | fprof96
```

The graphics output of **finteg96** is given in Figure 4.

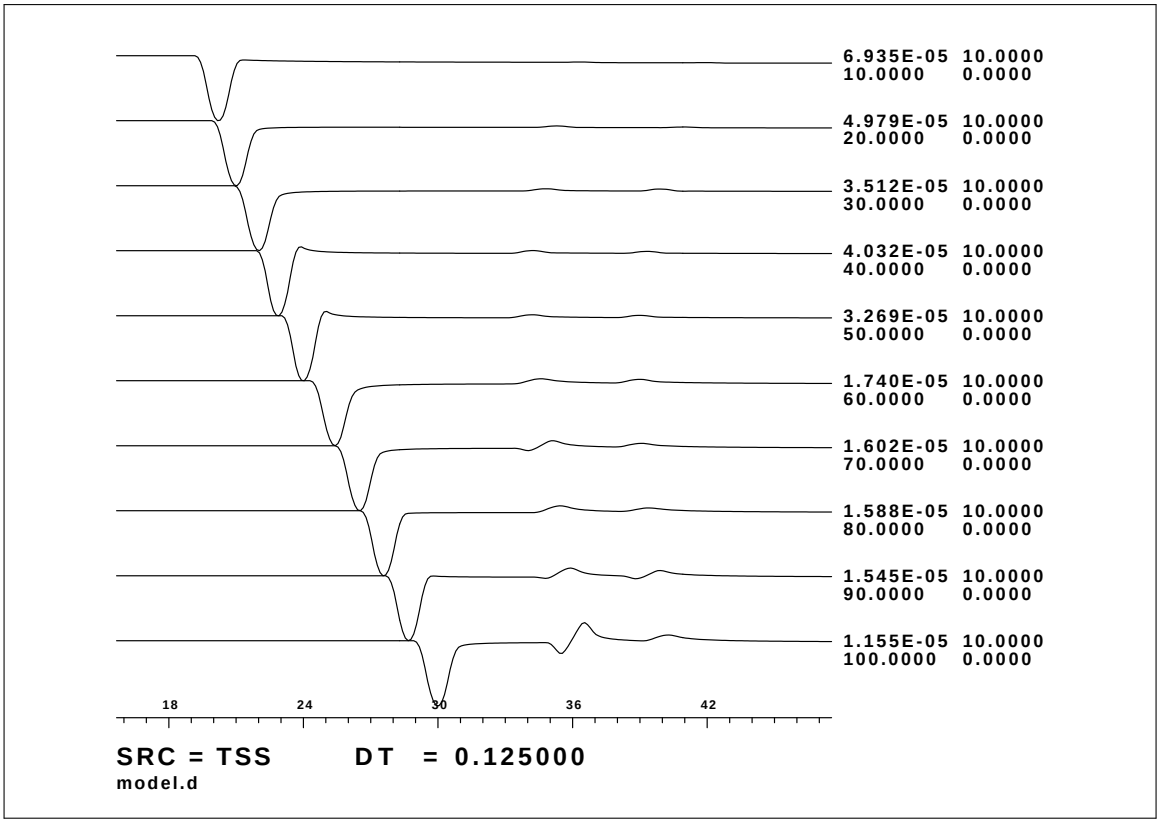
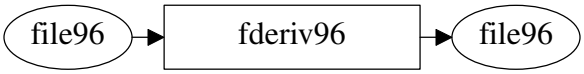


Fig. 4. Result of integrating the ground velocities.

6. fderiv96

This program differentiates a time history numerically. A simple backward difference is used. because of the simplicity of the algorithm, the **fderiv96 | finteg96** sequence will yield a one unit time delay. The algorithm implemented is

$$\left.\frac{dx}{dt}\right|_{t=n\Delta t} = \frac{x(n) - x(n-1)}{\Delta t}$$



Program control is through the command line:

fderiv96 *[flags]*, where the command flags are

-h

-?

Online help

Given the sample model and distance files of Chapter 1, the the generalized ray run of Chapter 2, following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV

rbh> genray96 -d dfile   > genray96.out

rbh> gpulse96 -V -p -l 4 > g.vel

rbh  fderiv96  < g.vel  | fprof96
```

The graphics output of **fderiv96** is given in Figure 5.

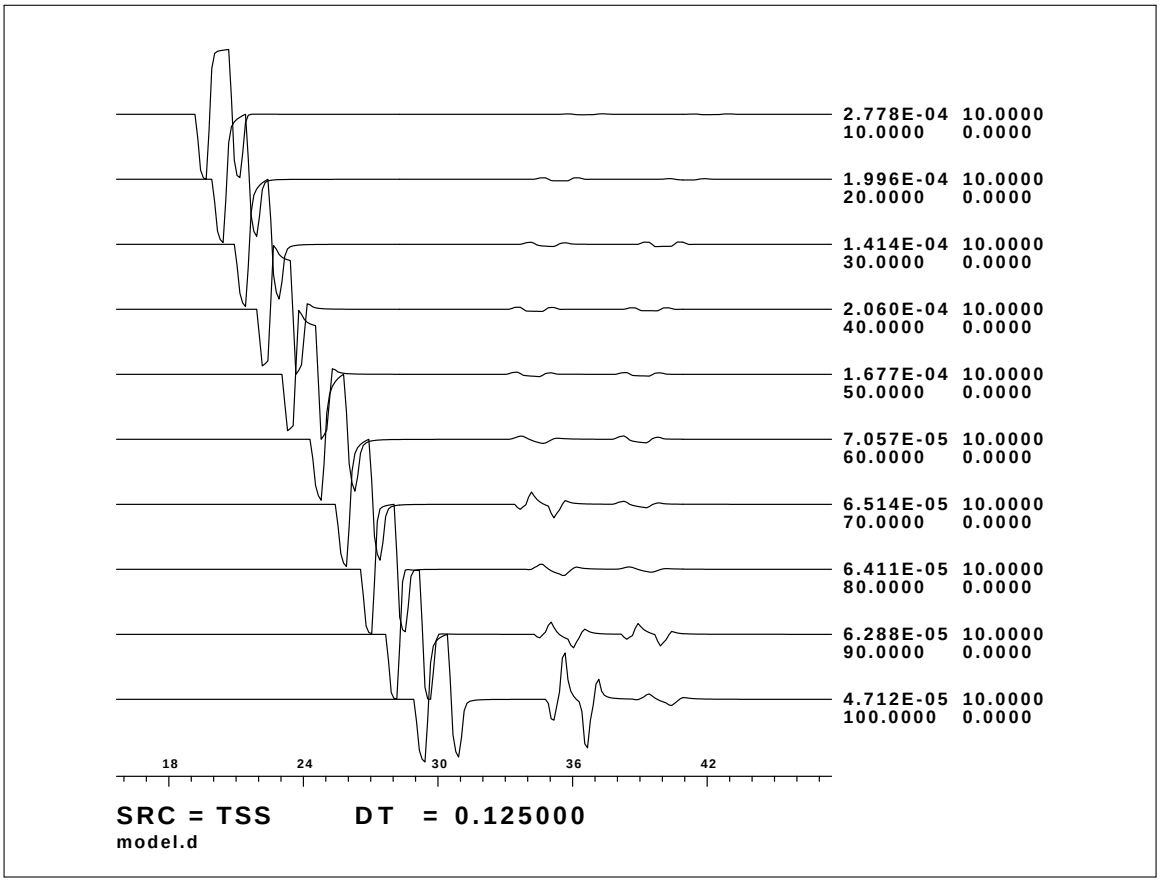
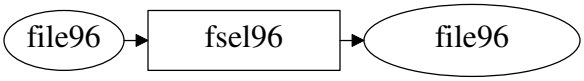


Fig. 5. Result of differentiating the ground velocity time history.

7. fsel96

This program selects subsets of a **file96(V)** to create a sequence of single trace **file96(V)** files.

Thus one may select all times histories (components or Green's functions) for one station or select a single component from all stations, e.g., to select all vertical components of three-component time histories. This may be useful when passing synthetic vertical component data through a phase velocity determination program. The processing flow is



Program control is through the command line:

fsel96 *[flags]*, where the command flags are

-NT number_trace Select only this trace/Green's function from each station. Note the program requires an integer, and thus it is required the the user know that 2 represents the RDD component for synthetic Green's functions or that 1 represents vertical component data for three component time histories.

-NS number_station Select all time histories for a single station/distance. The numbering is in the order of the sequence in the **file96** file.

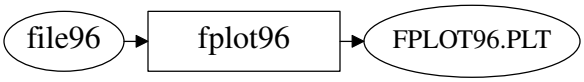
-h

-h

-? Command help

8. fplot96

This program creates the **CALPLOT(I)** file *FPLOT96.PLT* from the contents of the **file96(V)** file read from the standard input. For each station, e.g., each unique distance-depth set for synthetics, all traces are displayed on a single page. Subsequent pages show the results for other traces. The processing flow is



Program control is through the command line:

fplot96 *[flags]*, where the command flags are

-K kolor where **kolor** is an integer indicating CALPLOT color for the trace (default =1 which is black)

-T tmax (default all) maximum length of plot in sec

-A (default off) annotate with 1st arrival time

-h

-?

Online help

Given the sample model and distance files of Chapter 1, the the generalized ray run of Chapter 2, following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV
rbh> genray96 -d dfile   > genray96.out
rbh> gpulse96 -V -p -l 4 | fplot96
```

The first page of the graphics output of **fplot96** is given in Figure 6. This display may be of use to understand the individual ray contributions to the seismogram by analyzing the particle motion between the vertical and radial components for each Green's function.

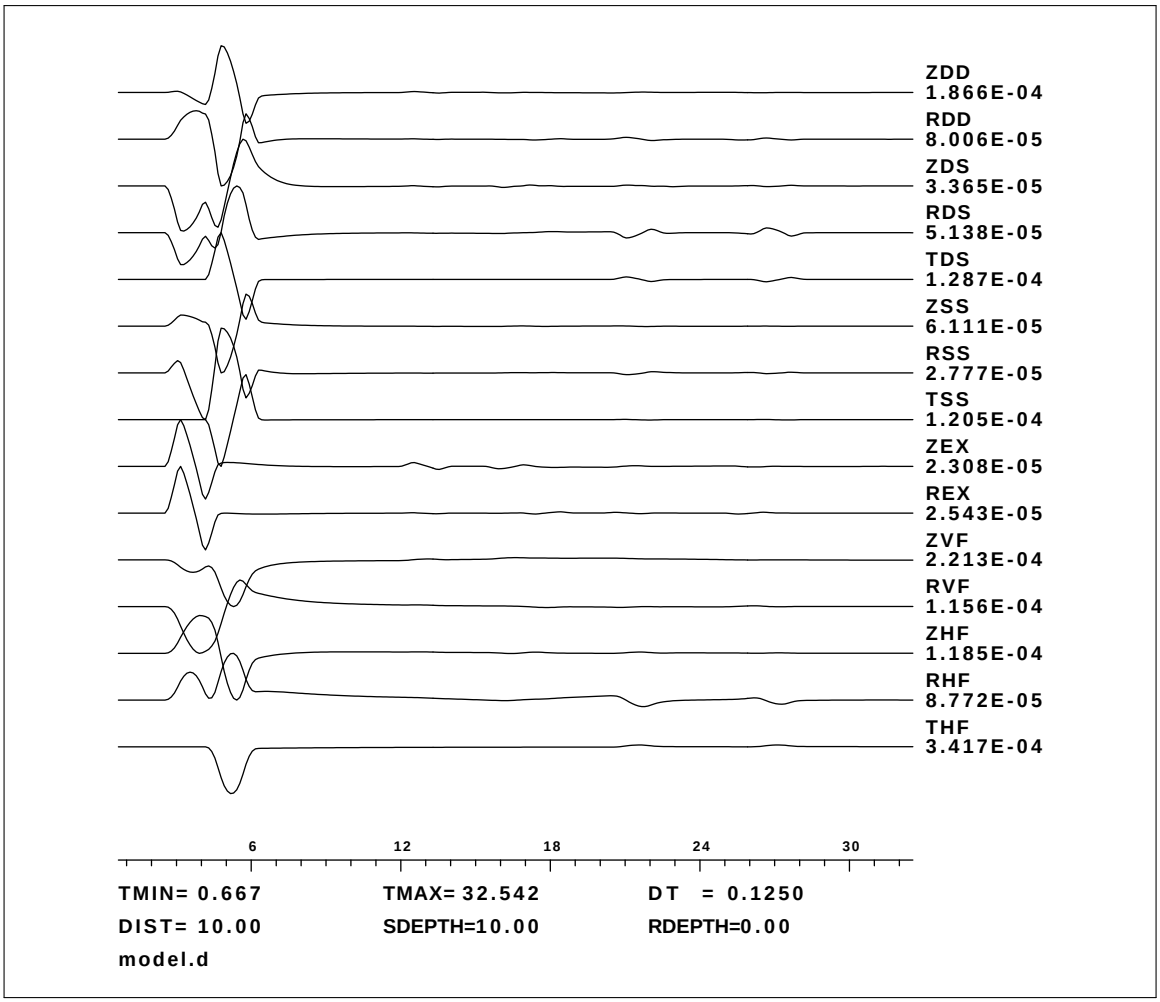
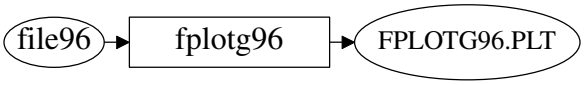


Fig. 6. The Green's functions for a distance of 10 kilometers.

9. fplotg96

This program creates the **CALPLOT(I)** file *FPLOTG96.PLT* from the contents of the **file96(V)** file read from the standard input. For each station, e.g., each unique distance-depth set for synthetics, all traces are displayed on a single page. Subsequent pages show

the results for other traces. The processing flow is



Program control is through command line flags:

fplotg96 *[flags]*, where the command flags are

-f1 file1 The name of the file to be plotted

-f2 file2 The name of the file to be plotted.

If neither flag is used, then the trace data is read from the standard input. If both are used, the traces are compared side by side.

-R

When both the **-f1 file1** and **-f2 file2** flags are used, the second trace is plotted using the scaling of the first trace, for each component. This provides direct comparison of amplitudes.

-K kolor where **kolor** is an integer indicating CALPLOT color for the trace (default =1 which is black)

-h

-?

Online help

Given the sample model and distance files of Chapter 1, and the generalized ray example of Chapter 2, following commands are run:

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV
rbh> genray96 -d dfile   > genray96.out
rbh> gpulse96 -V -p -l 4  > g.vel

rbh> hprep96  -M model.d -d dfile -HS 10 -HR 0 -ALL
rbh> hspec96   > hspec96.out
rbh> hpulse96 -V -p -l 4  > h.vel

rbh> fplotg -f1 g.vel -f2 h.vel -R
```

The first page of the graphics output of **fplotg96** is given in Figure 7.

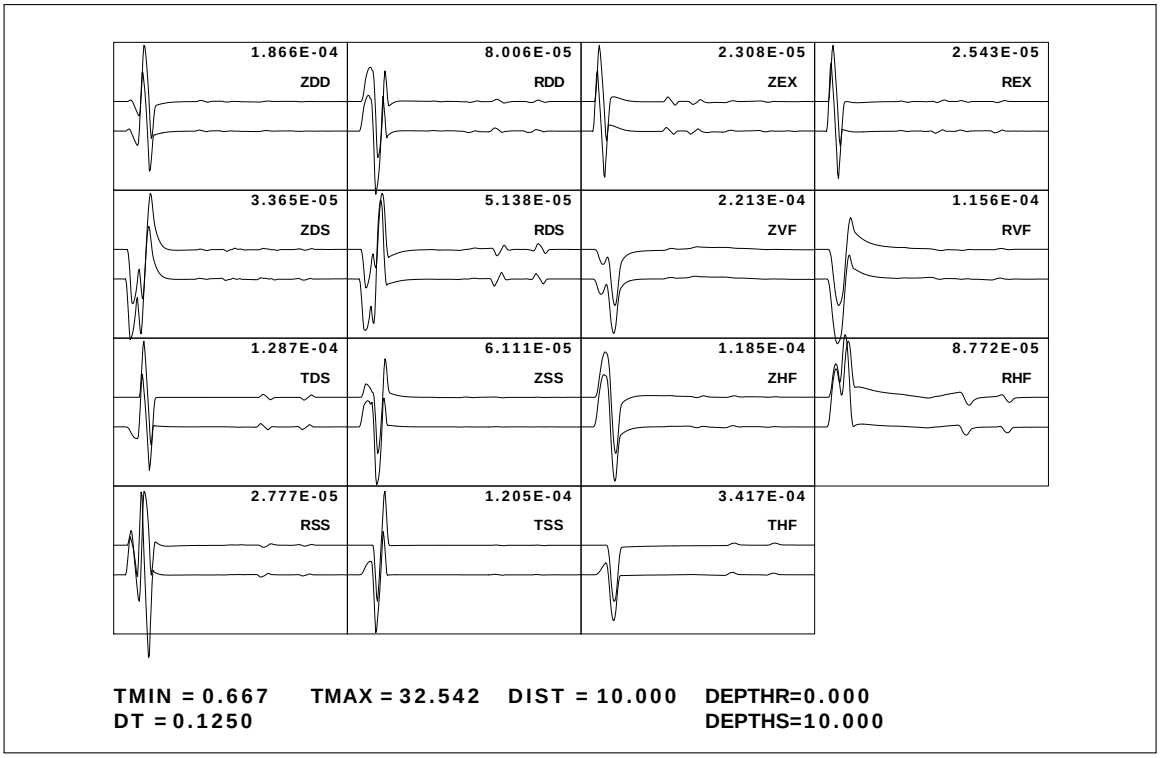
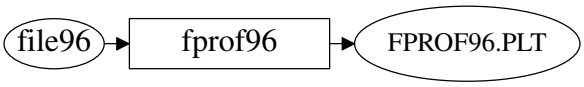


Fig. 7. The Green's functions for a distance of 10 kilometers, comparing generalized ray (upper trace) to the wavenumber integration (lower) for each Green's function.

10. fprof96

This program creates the **CALPLOT(I)** file *FPROF96.PLT* from the contents of the **file96(V)** file read from the standard input. This creates a unique page for each component. The processing flow is



Program control is through command line flags:

`fprof96` *[flags]*, where the command flags are

`-R`
The first trace of each component is used to define the scaling of subsequent traces on the same page.

`-K kolor` where **`kolor`** is an integer indicating CALPLOT color for the trace (default =1 which is black)

`-TMAX tmax` (default end of trace)

`-TMIN tmin` (default beginning of trace)
Define the time window for plotting the trace. This is relative to the beginning of the trace and not absolute time.

-S scaletrace (def 1.0) scale plotted trace factor
-A Annotate the trace with the P and S arrival times, if defined. (default do not annotate).
-h
-?
Online help

This program is used to create the figures used to describe **integ96** of this chapter.

11. fplot396

This program operates in a slightly different manner than the other plotting programs. Instead of directly plotting a **file96(V)** file in the manner of **fplot96(V)**, **fplotg96(V)**, or **fprof96(V)**, the list of files to be plotted is derived from a command file. A special command language is used. In addition, this program will only plot *three-component* time history files. The purpose is to be able to compare traces for publication.

The command file is read either from the standard input or is indicated on the command line. All other action is controlled by the program.

Program control is through command line flags:

fplot396 *[flags]*, where the command flags are
-C cmdfil The name of the command file. If not specified here, then it is derived from the standard input.
-h
-?
Online help

The command language used is as follows:

Command	Action
NEWPAGE	Force a new page
SHOWZRT	Place the symbols Z R T on the plot
SHOWZNE	Place the symbols Z N E on the plot If the file96 file only has Z R T, only Z is plotted. For side by side plots, these symbols are column headers.
NOSHOWZRT	Turn this option off
NOSHOWZNE	Turn this option off
RELATIVE	Top trace in column defines scaling. All other traces are relative to first
ABSOLUTE	Turn off the RELATIVE scaling
SHOWABC	Put sequential identifier to right of trace
RESETABC	Restart the sequencing at (a)
TIME	Put time axis in side by side plot at current position. Requires next line to have character string of times. *'tmin' 'tmax'*
SHOWTITLE	places title on right of trace. Next

	line is the title string, quoted. 'title'
SHOWFILE	places file name to right of trace.
NEWY0	Redefines position of top of plot. Default is 7.0 Next line is the floating point value *yy0*
DELY	Defines the vertical spacing between traces Default is 0.75. The Next line is the floating point value *dely*
FILE	Plot a file96 file. Next line contains name of file, start and end times of plotted segment with respect to origin time. File name is quoted. *'file_name' tstart tend*
VERTICAL	Plot traces vertically, not side by side
HORIZONTAL	Plot traces side by side (default)
NEWPEN	Next line gives integer indicating CALPLOT color. *kolor*

Two examples will illustrate the use of this program. The first will use the synthetic seismograms.

```
rbh> gprep96 -DOALL -HR 0 -HS 10 -M model.d -N 10  -DOCONV
rbh> genray96 -d dfile   > genray96.out
rbh> gpulse96 -V -p -l 4  > g.vel
rbh> fmech96 -D 45 -S 45 -R 45 -A 22.5 -B 202.5 < g.vel > zne.vel
rbh> fplot396 -C cmdfil
```

where the contents of the command file *cmdfil* are

```
SHOWZNE
SHOWABC
SHOWTITLE
 'SYN'
SHOWFILE
DELY
0.75
NEWPEN
4
HORIZONTAL
FILE
 'zne.vel' 0.0 80.0
TIME
 '0' '40'
```

Figure 8 shows the resulting plot. Note that plotting window is from 0 to 80 seconds after the origin time, but since the traces have different start times as a function of distance, they appear to shift to the right in the figure. If the data for individual distances had been in different files then the command word *TITLE* could have been used to indicate the receiver distances on the traces instead of using the (a) ... (j) tags.

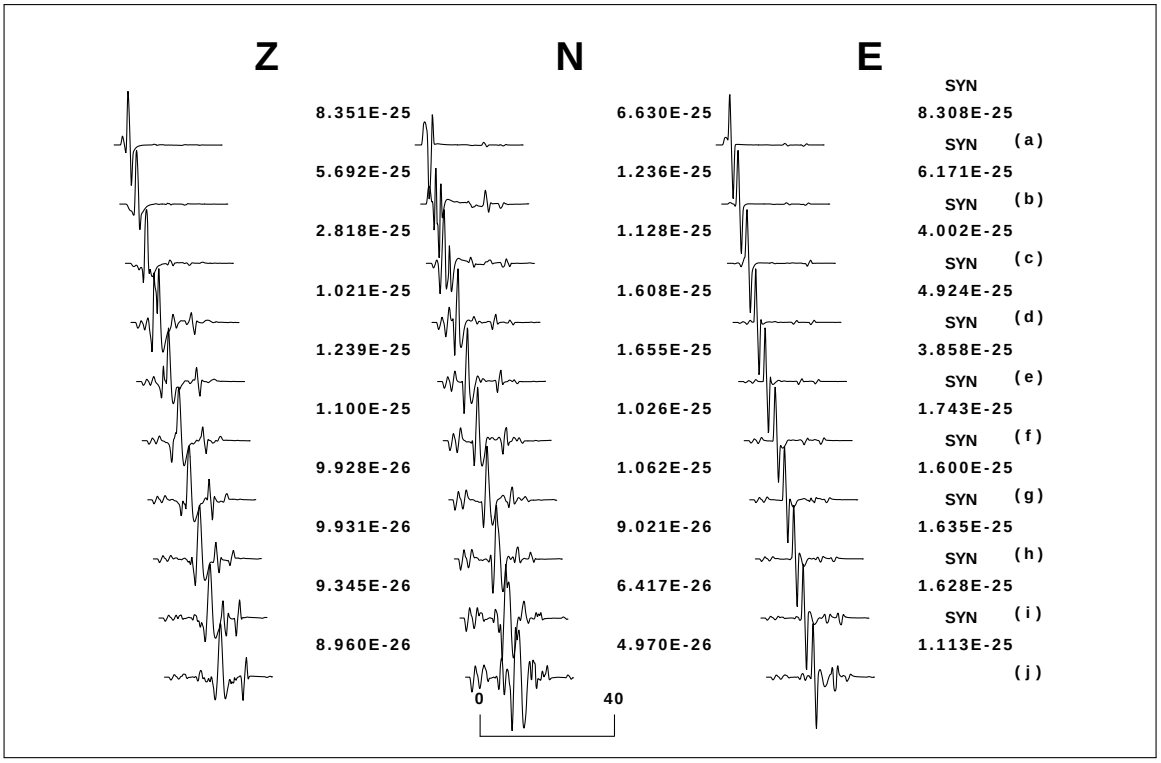


Fig. 8. The ground velocity as a function of distance. (a) is 10 km, (b) is 20 km, and (j) is 100 km.

The other example is from a synthetic seismogram fit to a real data set. In this case the command file is

```
SHOWZNE
SHOWABC
SHOWTITLE
 'FLO'
DELY
0.65
RESETABC
NEWPEN
2
RELATIVE
FILE
 '3f.wwlp.96' 20.0 120.0
SHOWTITLE
 '  '
FILE
 '3dataflo.196' 20.0 120.0
FILE
 '3dataflo.296' 20.0 120.0
FILE
 '3dataflo.396' 20.0 120.0
DOWN
0.65
RESETABC
RELATIVE
SHOWTITLE
 'SLM'
FILE
 '3s.wwlp.96' 20.0 120.0
SHOWTITLE
 '  '
FILE
 '3dataslm.196' 20.0 120.0
FILE
 '3dataslm.296' 20.0 120.0
FILE
 '3dataslm.396' 20.0 120.0
TIME
 '0' '30'
```

The resulting plot is given in Figure 9.

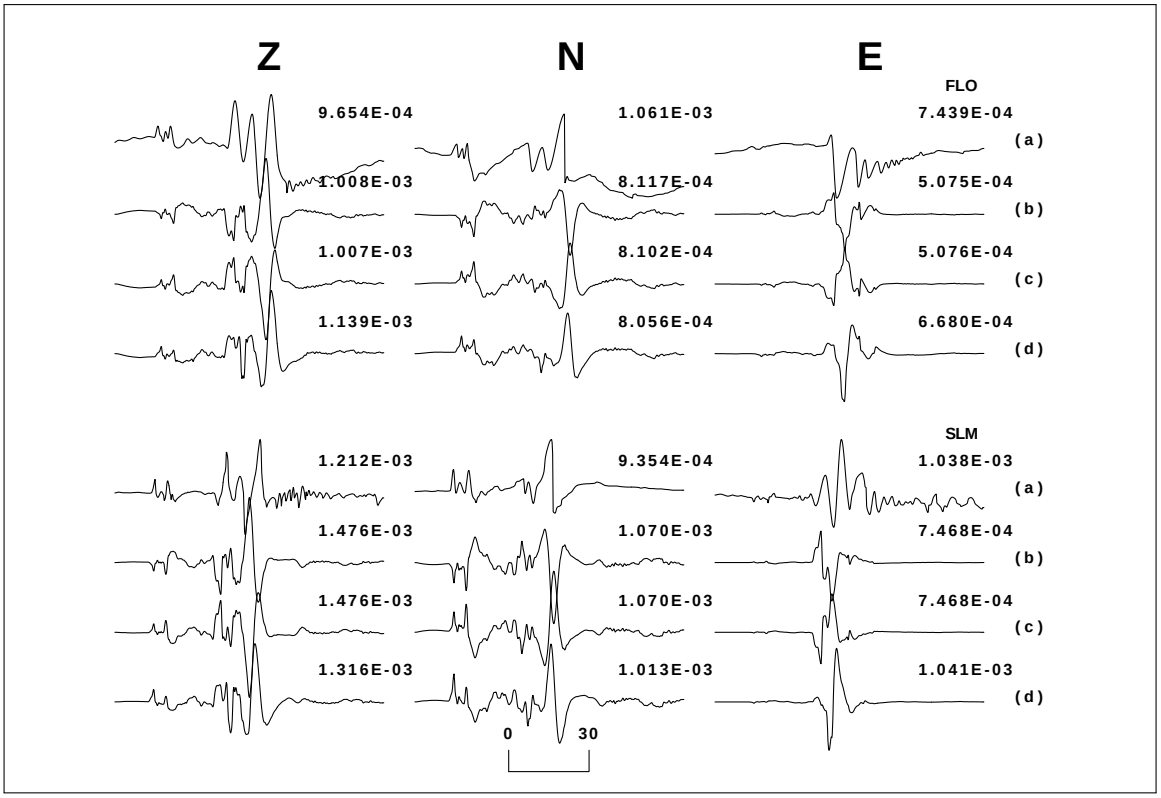


Fig. 9. Comparison of observed (a) and synthetic time histories for FLO and SLM for the March 3, 1963 Missouri earthquake. The traces are WWSSN 15-100 long period instrument with gain of 1.0. (b) if for the previously published mechanisms; (c) is that mechanism with P and T axes interchanged; (d) is that mechanism with P and T axes interchanged and rotated 180°. Trace amplitudes are in cm.

12. fspec96

Program control is through the command line:

fspec96 *[flags]*, where the command flags are
- **-K** *kolor* (default 1) pen color
- **-XLOG** (default .false.)
- **-XLIN** (default .true.)
- **-YLOG** (default .false.)
- **-YLIN** (default .true.)
- **-XMIN** *xmin* (default auto) minimum value of x-axis
- **-XMAX** *xmax* (default auto) maximum value of x-axis
- **-YMIN** *ymin* (default auto) minimum value of y-axis
- **-YMAX** *ymax* (default auto) maximum value of y-axis
- **-NX** *nx* (default auto) maximum cyles in log x-axis
- **-NY** *ny* (default auto) maximum cyles in log y-axis
- **-FREQ** (default .true.) x-axis is frequency
- **-PER** (default .false.) x-axis is period
- **-?**

-h Online help

13. fdecon96

fdecon96 performs a deconvolution in the frequency domain by dividing the spectra of the numerator by the spectra of the denominator. To bandlimit the results for a successful deconvolution, a cosine taper can applied. The deconvolution uses a water level set as 0.001 maximum spectral amplitude of the denominator.

The program is invoked as

fdecon96 *[flags]*
-FN file_num (default none) numerator
-FD file_den (default none) denominator
-W (default false) force cosine taper in freq domain
-?
-h Online help.

A sample invocation would be

```
fdecon96 -N numerator -D denominator
```

The output is given in the standard output.

CHAPTER 7
SAC FILTERS REVISED

1. Introduction

This chapter describes programs written to interact with SAC files. **SAC - Seismic Analysis Code** is a seismic trace analysis package written at Lawrence Livermore National Laboratory and is copyrighted by the Regents of the University of California. At present SAC runs on a number of different hardware and operating system platforms, MS-DOS and Apple Macintosh excepted. While the package is powerful, there is often a need to convert SAC binary trace files to other formats without using the actual program.

The synthetic seismogram code presented here uses a **file96(V)** format that differs from the SAC trace format by permitting multiple components and multiple stations on the same ASCII file. Thus there is a need to quickly convert between SAC and the **file96(V)** format. This chapter presents programs that work with SAC files.

2. sactoasc

This program converts a SAC binary trace file to a SAC ascii file. The command is run directly from the command line as:

```
sactoasc SAC_BINARY_FILE SAC_ASCII_FILE
```

The order of arguments is important. The *SAC_ASCII_FILE* will be over-written if it already exists.

3. asctosac

This program converts a SAC ascii trace file to a SAC binary trace file. The command is run directly from the command line as:

```
asctosac SAC_ASCII_FILE SAC_BINARY_FILE
```

The order of arguments is important. The *SAC_BINARY_FILE* will be over-written if it already exists.

4. shwsac

This program reads a SAC file given on the command line, and tells everything possible about the contents of the SAC header. In addition, a simple plot is presented of the trace. The command is run with arguments on the command line:

```
shwsac [-A] [-B] SAC_FILE
```

where the flags **[-A]** and **[-B]** indicate that the SAC_FILE is ascii or binary, respectively. The user is responsible for specifying the type of SAC file, since this information is not directly obtainable from the file itself. An example of the output follows from the invocation of the command

```
shwsac -B B0101ZDD.sac
```

for which the SAC file was created by **f96osac(V)**. The screen output is

```
   REAL         INDEX  NAME         INT VALUE   REAL VALUE
                    1  DELTA     0.12500E+00  0.12500E+00
                    2  DEPMIN   -0.86318E-05 -0.86318E-05
                    3  DEPMAX    0.13413E-04  0.13413E-04
                    4  SCALE    -0.12345E+05 -0.12345E+05
                    5  ODELTA   -0.12345E+05 -0.12345E+05
                    6  B         0.00000E+00  0.00000E+00
                    7  E         0.31875E+02  0.31875E+02
                    8  O         0.00000E+00  0.00000E+00
                    9  A        -0.12345E+05 -0.12345E+05
                   10  FMT      -0.12345E+05 -0.12345E+05
                   11  T0       -0.12345E+05 -0.12345E+05
                   12  T1       -0.12345E+05 -0.12345E+05
                   13  T2       -0.12345E+05 -0.12345E+05
                   14  T3       -0.12345E+05 -0.12345E+05
                   15  T4       -0.12345E+05 -0.12345E+05
                   16  T5       -0.12345E+05 -0.12345E+05
                   17  T6       -0.12345E+05 -0.12345E+05
                   18  T7       -0.12345E+05 -0.12345E+05
                   19  T8       -0.12345E+05 -0.12345E+05
                   20  T9       -0.12345E+05 -0.12345E+05
                   21  F        -0.12345E+05 -0.12345E+05
                   22  RESP0    -0.12345E+05 -0.12345E+05
                   23  RESP1    -0.12345E+05 -0.12345E+05
                   24  RESP2    -0.12345E+05 -0.12345E+05
                   25  RESP3    -0.12345E+05 -0.12345E+05
                   26  RESP4    -0.12345E+05 -0.12345E+05
                   27  RESP5    -0.12345E+05 -0.12345E+05
                   28  RESP6    -0.12345E+05 -0.12345E+05
                   29  RESP7    -0.12345E+05 -0.12345E+05
                   30  RESP8    -0.12345E+05 -0.12345E+05
                   31  RESP9    -0.12345E+05 -0.12345E+05
                   32  STLA      0.00000E+00  0.00000E+00
                   33  STLO      0.00000E+00  0.00000E+00
                   34  STEL      0.00000E+00  0.00000E+00
                   35  STDP     -0.12345E+05 -0.12345E+05
                   36  EVLA      0.00000E+00  0.00000E+00
                   37  EVLO      0.10000E+02  0.10000E+02
                   38  EVEL     -0.12345E+05 -0.12345E+05
                   39  EVDP      0.00000E+00  0.00000E+00
```

```
                  40  FHDR40    -0.12345E+05 -0.12345E+05
                  41  USER0     -0.12345E+05 -0.12345E+05
                  42  USER1     -0.12345E+05 -0.12345E+05
                  43  USER2     -0.12345E+05 -0.12345E+05
                  44  USER3     -0.12345E+05 -0.12345E+05
                  45  USER4     -0.12345E+05 -0.12345E+05
                  46  USER5     -0.12345E+05 -0.12345E+05
                  47  USER6     -0.12345E+05 -0.12345E+05
                  48  USER7     -0.12345E+05 -0.12345E+05
                  49  USER8     -0.12345E+05 -0.12345E+05
                  50  USER9     -0.12345E+05 -0.12345E+05
                  51  DIST       0.89930E+00  0.89930E+00
                  52  AZ         0.18000E+03  0.18000E+03
                  53  BAZ        0.00000E+00  0.00000E+00
                  54  GCARC      0.00000E+00  0.00000E+00
                  55  SB        -0.12345E+05 -0.12345E+05
                  56  SDELTA    -0.12345E+05 -0.12345E+05
                  57  DEPMEN    -0.72922E-08 -0.72922E-08
                  58  CMPAZ      0.00000E+00  0.00000E+00
                  59  CMPINC     0.00000E+00  0.00000E+00
                  60  XMINIMUM  -0.12345E+05 -0.12345E+05
                  61  XMAXIMUM  -0.12345E+05 -0.12345E+05
                  62  YMINIMUM  -0.12345E+05 -0.12345E+05
                  63  YMAXIMUM  -0.12345E+05 -0.12345E+05
                  64  ADJTM     -0.12345E+05 -0.12345E+05
                  65  FHDR65    -0.12345E+05 -0.12345E+05
                  66  FHDR66    -0.12345E+05 -0.12345E+05
                  67  FHDR67    -0.12345E+05 -0.12345E+05
                  68  FHDR68    -0.12345E+05 -0.12345E+05
                  69  FHDR69    -0.12345E+05 -0.12345E+05
                  70  FHDR70    -0.12345E+05 -0.12345E+05
INTEGER       INDEX  NAME          INT VALUE     INT VALUE
                   1  NZYEAR             1970          1970
                   2  NZJDAY                1             1
                   3  NZHOUR                0             0
                   4  NZMIN                 0             0
                   5  NZSEC                15            15
                   6  NZMSEC              666           666
                   7  NVHDR                 6             6
                   8  NINF                  0             0
                   9  NHST                  0             0
                  10  NPTS                256           256
                  11  NSNPTS           -12345        -12345
                  12  NSN              -12345        -12345
                  13  NXSIZE           -12345        -12345
                  14  NYSIZE           -12345        -12345
                  15  NHDR15           -12345        -12345
ENNUMERATED   INDEX  NAME          INT VALUE     ENU VALUE
                  16  IFTYPE                1      ITIME
                  17  IDEP             -12345
                  18  IZTYPE                9      IB
                  19  IHDR4            -12345
                  20  IINST            -12345
                  21  ISTREG           -12345
                  22  IEVREG           -12345
                  23  IEVTYP           -12345
                  24  IQUAL            -12345
                  25  ISYNTH           -12345
LOGICAL       INDEX  NAME          INT VALUE     LOG VALUE
                  36  LEVEN                 1             T
                  37  LPSPOL                0             F
                  38  LOVROK                0             F
                  39  LCALDA                0             F
                  40  LHDR5                 0             F
CHARACTER     INDEX  NAME          INT VALUE    CHAR VALUE
```

```
                 1  KSTNM         GRN16         GRN16
                 2  KEVNM         SYNTHETI      SYNTHETI
                 3  KEVNMC        C             C
                 4  KHOLE         -12345        -12345
                 5  KO            -12345        -12345
                 6  KA            -12345        -12345
                 7  KT0           -12345        -12345
                 8  KT1           -12345        -12345
                 9  KT2           -12345        -12345
                10  KT3           -12345        -12345
                11  KT4           -12345        -12345
                12  KT5           -12345        -12345
                13  KT6           -12345        -12345
                14  KT7           -12345        -12345
                15  KT8           -12345        -12345
                16  KT9           -12345        -12345
                17  KF            -12345        -12345
                18  KUSER0        -12345        -12345
                19  KUSER1        -12345        -12345
                20  KUSER2        -12345        -12345
                21  KCMPNM        ZDD           ZDD
                22  KNETWK        -12345        -12345
                23  KDATRD        -12345        -12345
                24  KINST         -12345        -12345
1  0.
26    7.40105E-07
51    1.51866E-07
76    8.95409E-08
101    5.38562E-07
126   -8.63175E-06
151   -2.72976E-06
176   -2.86479E-07
201   -2.29318E-09
226    1.70056E-08
251    1.41848E-08
256    1.33739E-08
```

and the graphics output is shown in Figure 1. The table contains the components of the floating point, integer and character headers of the SAC file. In addition, the integer values may be interpreted as logical or to represent some ennumerated values. The second column shows the SAC internal name, or something like FHDR70 if there is not official SAC name, the actual value in the header, and the enumerated value. Finally the last 12 lines show the first, last and ten intermediate values of the time history.

5. f96tosac Updated

This program converts a **file96(V)** trace file to single trace SAC files, either in binary or ASCII format. The program is controlled from the command line as follows:

`f96tosac` *[flags]*

`OUTPUT FILE NAME`
`-A` SAC alphanumeric file, else binary
`-B` SAC binary (default)
`-G` (default) Output names in form DDDDdHHHh.grn (binary)

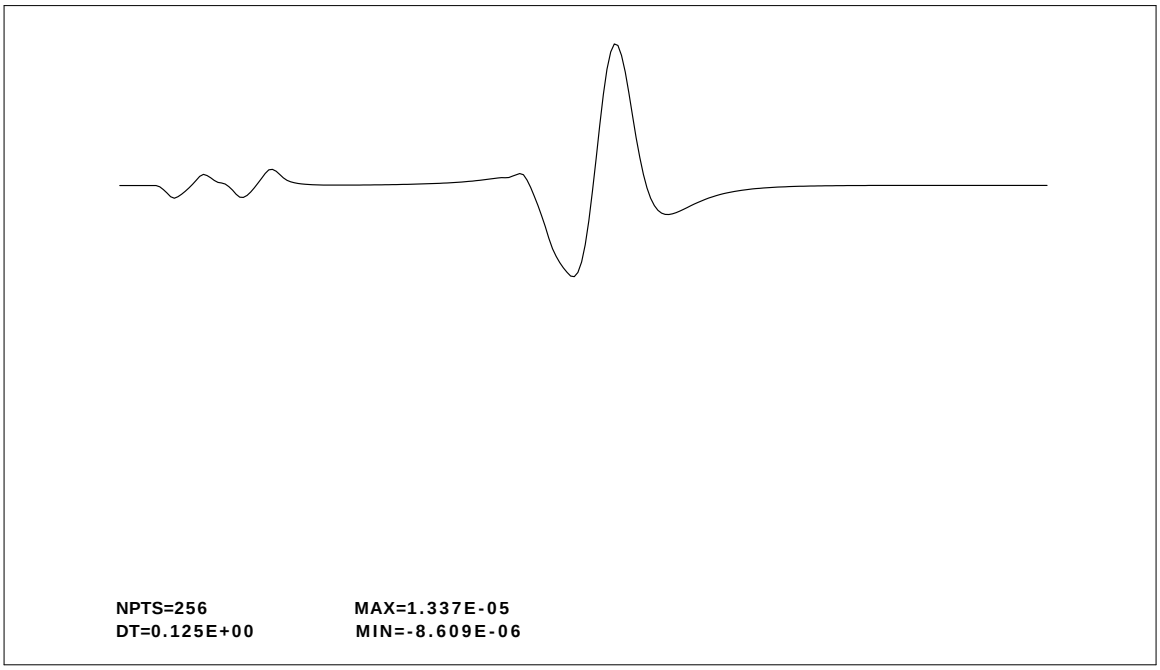


Fig. 1. Graphic output of the program **shwsac(V)**.

-T Output names in form DDDDDdHHHh.grn(binary)
-E Output names in form DDDdddHhhh.grn(binary)
The format for the name of the binary output attempts to
give information on epicentral distance (km),
source depth (km), and receiver depth(km). The options are
-FMT 1 DDDDDd_HHHh_ZZZz.grn e.g. 005001_1234_0045.Uz
-FMT 2 DDDDDddd_HHHhhh_ZZZzzz.grn e.g. 00500123_123456_004578.ZVF
-FMT 3 DDDDDdHHHh.grn (same as -T) e.g. 0050010041.ZVF
-FMT 4 DDDDdHHHh.grn (same as -G) e.g. 050010045.ZVF
-FMT 5 DDDdddHhhh.grn (same as -E) e.g. 5001234578.ZVF
where D is for epicentral distance, H source depth, and
Z receiver depth. The lower case indicates the digits
to the right of the decimal place. The examples above
are for an epicentral distance is 500.123 km, source
depth 123.456 km and receiver depth 4.578 km.
The -FMT x naming convention is compatible with the
mt command of gsac. Howver the model must be in units of
km, km/s and g/cm^3
-? This online help
-h This online help

The many options are provided for the convenience of the user. Originally there was interest only in the Green's functions as a funciton of distance. Later, there was a need to view synthetics for different receiver depths. The only limitation occurs when the rnange of distance, source and reciever depth cannot be described by the FORTRAN format fields. Thus there would be problems if the receiver depth is 0.001 km, in which cse the ZZZz woould be 0000. One can always change the source code.

The naming of the GREEN's functions is discussed in Appedix B.

6. sactof96

The conversion of SAC files to the **file96(V)** format is not trivial, since all files for one station must be input. The program **sacfto96** uses an interactive command language to accomplish this. The program will create a **file96(V)** file, with the fixed name *file96.fil*, for only one station.

The command line language consists of single line commands, which can be in either upper or lower case, and optional second line entries. Only the commands *UNIT* or those indicating a specific component, e.g., *01* or *Z* are followed by an additional line. The command syntax can be seen by entering the command *HELP* which prompts the user for an *enter* so the screen does not fill so quickly. The results of the *HELP* command are:

```
Control the creation of FILE96(V) files by these
commands:
KEYWORD            MEANING
_____________________________________________________
HELP               Print this help screen
REVIEW             Print current input list
BINARY             SAC file is in binary form
ASCII              SAC file is in ASCII form
OBSERVED           Series is observed
SYNTHETIC          Series is synthetic
TIME_DOMAIN        This is a time series
FREQUENCY_DOMAIN   This is a complex Fourier transform series
PROCESS            Begin conversion to FILE96
QUIT               Exit this program
FILE01             Create a  1 component FILE96
FILE03             Create a  3 component FILE96
FILE16             Create a 16 component FILE96
UNIT               The next line is one of the
                   following left justified keywords:
       COUNTS CM CM/SEC CM/SEC/SEC M M/SEC
       M/SEC/SEC MICRON MICRON/SEC MICRON/SEC/SEC
01        The following FILE is ZDD
02        The following FILE is RDD
03        The following FILE is ZDS
04        The following FILE is RDS
05        The following FILE is TDS
06        The following FILE is ZSS
07        The following FILE is RSS
08        The following FILE is TSS
09        The following FILE is ZEX
10        The following FILE is REX
11        The following FILE is ZVF
12        The following FILE is RVF
13        The following FILE is ZHF
14        The following FILE is RHF
15        The following FILE is THF
Z         The following FILE is Vertical
N         The following FILE is North
E         The following FILE is East
T         The following FILE is Transverse
O         The following FILE is Other
_____________________________________________________
ENTER to continue
```

An actual run combining three of the SAC files created by **f96tosac** into a three component file required the following commands:

```
FILE03
SYNTHETIC
UNIT
CM/SEC
BINARY
Z
B0106ZSS.sac
R
B0107RSS.sac
T
B0108TSS.sac
REVIEW
PROCESS
```

The *REVIEW COMMAND* provides the current state of the desired conversion. The results for this run are:

```
        File Type:       FILE03
        Data Source:     SYNTHETIC
        Domain:          TIME_DOMAIN
        Units:           CM/SEC
        SAC File Type:   BINARY
  Z     SAC Data File:   B0106ZSS.sac -- Exists
  R     SAC Data File:   B0107RSS.sac -- Exists
  T     SAC Data File:   B0108TSS.sac -- Exists
```

Figure 2 shows the plot of the contents of the file *file96.fil* created using the command **`fplot96 < file96.fil`**.

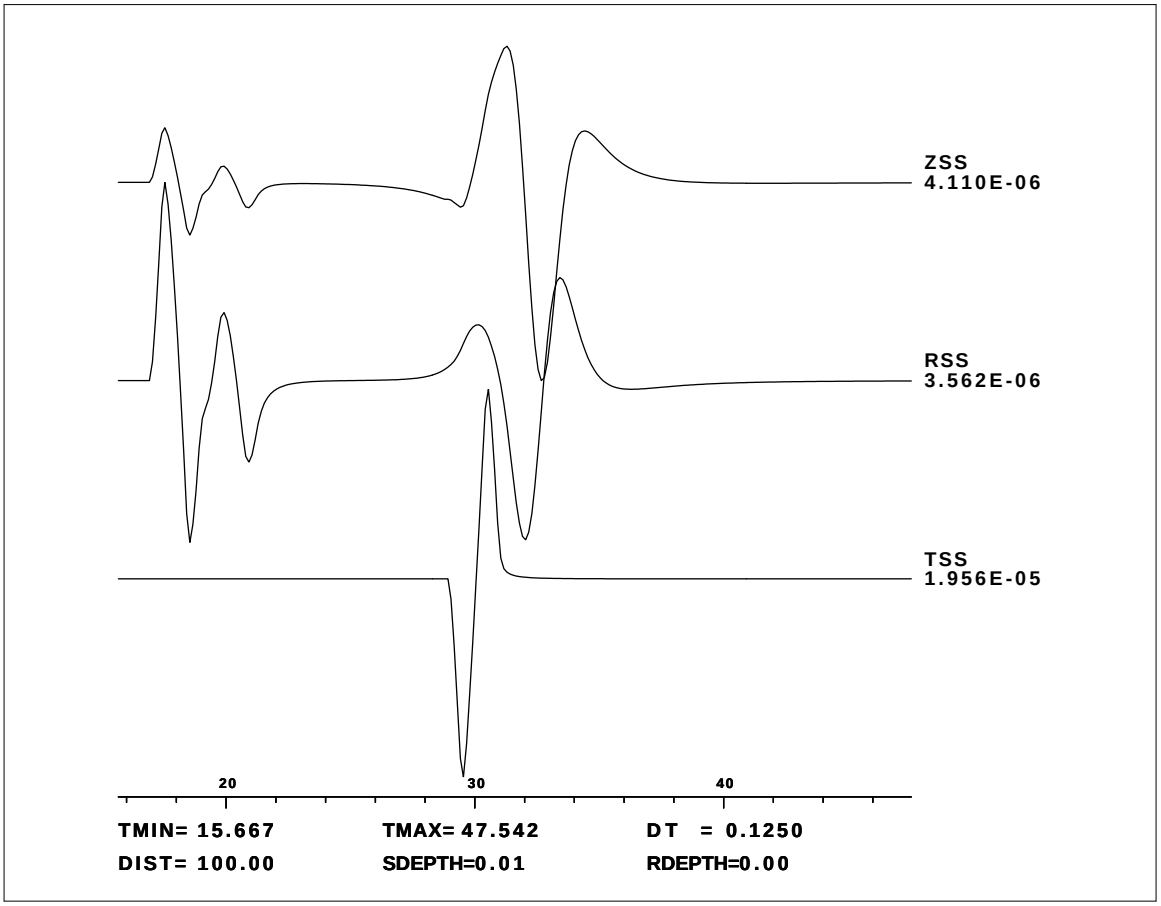


Fig. 2. Plot of the **file96** file created by **sactof96**.

7. sacdecon

sacdecon performs a deconvolution in the frequency domain by dividing the spectra of the numerator by the spectra of the denominator. To bandlimit the results for a successful deconvolution, a cosine taper can applied. The deconvolution uses a water level set as 0.001 maximum spectral amplitude of the denominator.

The program is invoked as

sacdecon *[flags]*
- **-FN** *file_num* (default none) numerator
- **-FD** *file_den* (default none) denominator
- **-W** (default 0.001) water level
- **-T** (default false) force cosine taper in freq domain
- **-ALP** *alpha* (default 2.3) complex frequency parameter
- **-D** *delay* (0 sec) Begin output delay sec before t=0
- **-A** (default false) data are SAC ascii
- **-B** (default true) data are SAC binary
- **-?**
- **-h** Online help.

A sample invocation would be

```
sacdecon -FN numerator -FD denominator -B
```

The output is given in the file *sacdecon.bin* if the original files were in SAC binary and in the file *sacdecon.asc* if the original files were in SAC alpha format.

8. saciterd

saciterd performs a time domain deconvolution. This program was written by C. J. Ammon and is described in

> Ligorria, J. P. and C. J. Ammon (1999). Iterative deconvolution and receiver-function estimation, *Bull. Seism. Soc. Am.* **89,** 1395-1400.

An important aspect of this program is that the deconvolution is expressed as a sequence of Gaussian filtered impulses. The zero phase Gaussian filter is defined as

$$\mathbf{H(f) = e^{-\pi^2 f^2/\alpha^2}} .$$

The parameter alpha controls the frequency content, with the e^{-1} point at a frequency α/π. Thus an $\alpha = 1.0$ give a lowpass version of the receiver function at a frequency of about 0.3 Hz.
The program is invoked as

saciterd *[flags]*
- **-FN** *file_num* (default none) numerator
- **-FD** *file_den* (default none) denominator
- **-E** *error* (0.001) convergence criteria
- **-ALP** *alpha* (default 1.0) Gaussian Filter Width
  H(f) = exp(- (pi freq/alpha)**2)
  Filter corner ~ alpha/pi
- **-N** *niter* (default 100) Number iterations/bumps
- **-D** *delay* (5 sec) Begin output delay sec before t=0
- **-POS** (default false) Only permit positive amplitudes
- **-2** (default false) use double length FFT to avoid FFT wrap around in convolution
- **-RAYP** *rayp* (default -12345.0) Ray parameter in (sec/km) to set in SAC header for use by **rftn96** and **joint96**. This value is not used by this program. Use **udtdd** to determine this value
- **-?**
- **-h** Online help.

Output files:
- *observed* : original numerator convolved with Gaussian
- *numerator* : original numerator convolved with Gaussian
- *denominator* : original numerator convolved with Gaussian
- *decon.out* : Receiver function for Gaussian
- *predicted* : Receiver function for Gaussian

SAC header values set in *decon.out*:
SAC header values set"
B : delay
USERO : gwidth KUSER0: Rftn
USER4 : rayp (sec/km)
USER5 : fit in %
KEVNM : Rftn KUSER1: IT_DECON
USER0 : gwidth USER5 : fit in %

A sample invocation would be

```
saciterd -FN file_num -FD file_den
```

where *numerator* and *denominator* are binary SAC files.

This program creates several files which are in SAC binary format:
decon.out - final Gaussian filtered deconvolution
observed - Gaussian filtered numerator
numerator - Gaussian filtered numerator
denominator - Gaussian filtered denominator
predicted - prediction of Gaussian filtered numerator obtained by convolving *decon.out* with the original denominator fil *file_den*

In addition the program sets several of the SAC header values in the file *decon.out*:

B - set to *-delay* seconds
USER0 - set to the value of the Gaussian filter parameter *alpha*
KUSER0 - set to the string *Rftn*.
KUSER1 - set to the string *IT_DECON*.
USER4 - set to the value of the ray parameter (sec/km) on the command line. If not specified on the command line, the SAC default value of -12345.0 is used.
KUSER4 - if the ray parameter > 0, then this variable is set to *p(km/s)*.
USER5 - set to the quality of fit value, e.g., for the example below this will be *99.5* indicating that the predicted numerator accounts for 99.5% of the actual filtered numerator.
KUSER5 - set to the string *FIT*

The following example uses the program **`hspec96p`** to generate vertical and radial plane-wave synthetics for a teleseismic signal incident from a halfspace onto a 1 km thick soil deposit. The vertical and radial binary SAC files are denoted as `file.Z` and `file.R`, respectively. **`saciterd`** is invoked to perform 100 iterations with α = **5**.

Figure 3 presents the input traces and the traces generated by the script:

```
#!/bin/sh

cat > dfile << EOF
100.0 0.01 4096  5.0 0.0
EOF

#####
#               define the simple soil model
#####

cat > soil.mod << EOF
MODEL
simple soil model
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
HR      VP      VS   RHO QP   QS   ETAP ETAS FREFP FREFS
1.00     1.80   0.80 1.6 100 100 0.0 0.0 1.0 1.0
10.0     6.1000 3.55 2.7 100 100 0.0 0.0 1.0 1.0
EOF

#####
#               Make synthetic plan wave response
#               This should be a good approximation for teleseisms
#               especially since we are interested in the receiver function
#
#               We only consider upgoing P saves form the source in hspec96p
#####

hprep96p -M soil.mod -d dfile -HS 10 -HR 0 -TF -BH -EQEX -PMIN 0.07 -PMAX 0.07 -DF 0.0
hspec96p -SPUP
hpulse96 -p -V -l 2 | f96tosac -B
mv B0109ZEX.sac file.Z
mv B0110REX.sac file.R
rm B*sac

#####
#               now run saciterd
#####

saciterd  -FN file.R -FD file.Z -N 100 -D 10.0 -E 0.00001 -ALP 5.0 -RAYP 0.07
```

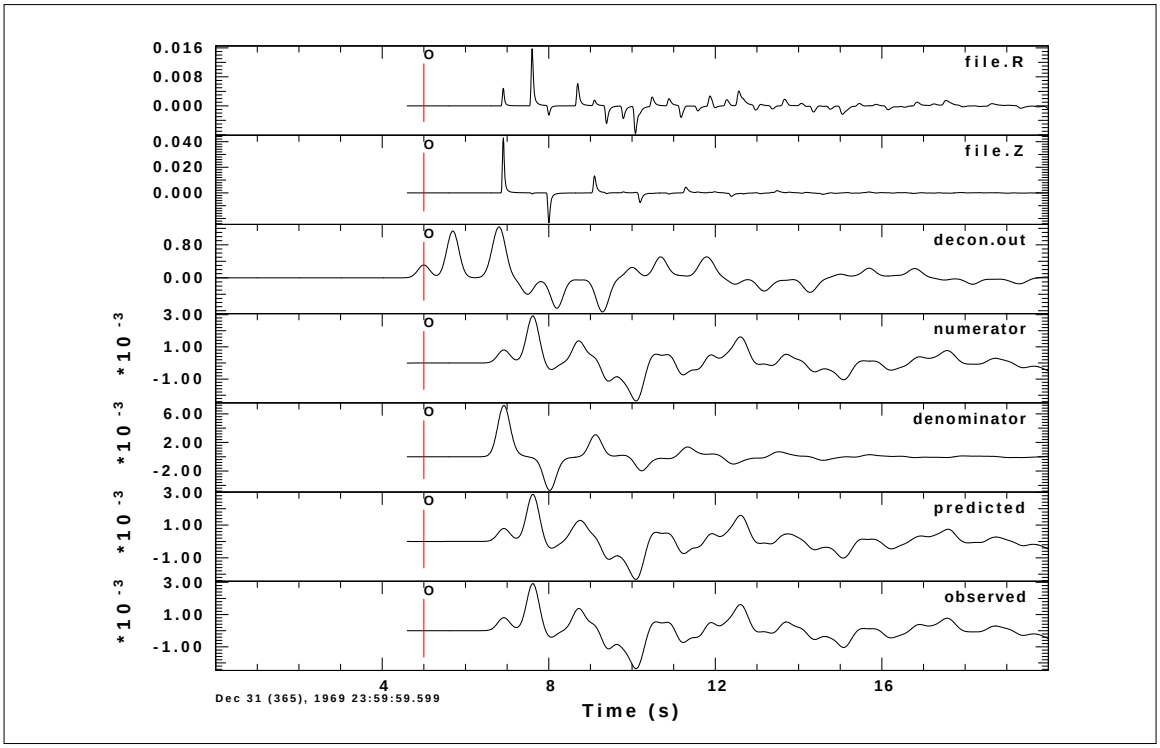


Fig. 3. Traces in order from top to bottom: `file.R`, radial time series; `file.Z`, vertical time series; `decon.out`, receiver function; `numerator`; `denominator`; `predicted` and `observed`. The file `decon.out` would be input to a program to invert the receiver funciton for earth structure.

If the **-V** flag is invoked, SAC files are created at each stage of the iteration with names `rXXX`, `dXXX` and `fXXX`, where XXX is the three digit iteration number. For example, `p001` is the predicted Gaussian filtered numerator at iteration 1, `d001` is the receiver function as of iteration 1, and `r001` is the residual receiver function to be fit by further iterations.

9. sacevalr

Rather than using this with the *file96* files, use the **GSAC** *transfer command* with the arguments **eval subtye amp_file phase_file**.

Using the amplitude and phase ascii files output from the IRIS program **evalresp**, this program convolves or deconvolves the instrument response.

sacevalr *[flags]*

-DEMEAN (default false) remove mean before filter
-TAPER (default false) apply taper before filter
-FREQLIMITS f1 f2 f3 f4 apply a taper for deconvolution (-R) This cubic taper ensures that the deconvoled spectrum passband is [f2,f3] and that the spectum is zero for **f < f1** and **f > f4** with a cubic taper int the region [f1,f2] and [f3,f4].

-A (default true) apply filter
-R (default false) remove filter
-AMP *amp_file* (none) evalresp amp file
-PHA *phase_file* (none) evalresp phase file
-SACIN *binary_sac_input* file (none)
-SACOUT *binary_sac_output* file (none)
-?
-h
Online help

In addition the program sets several of the SAC header values in the file *binary_sac.out* if the instrument response is removed:

USER1 - minimum period in the passband
USER2 - maximum period in the passband
KUSER1 - set to the string *PER_MIN*.
KUSER2 - set to the string *PER_MAX*.
These fields are set when removing the instrument response (**-R**) so that other programs, such as **sacmft96** and **sacpom96**, only use frequencies within the passband of deconvolution. The use of frequency limits ensures a stable deconvolution by a zero-phase bandbass.

10. saclhdr

This is a very useful program to interrogate the values in the header of a sacfile and then to use those values in a shell script. This is how **gsac** can fundtion without the use of blackboard variables.

This program examines the header of a SAC file to return a specified header value. The purpose of the program is to return the header value in a way that permits is to be assigned to a SHELL variable. The following shell script illustrates its use. Note that a different syntax is required in *csh*.

```
#!/bin/sh
#####
#       test sac file using the new saclhdr
#####

ISSAC=`saclhdr -ISSAC $1`

if test $ISSAC = 1
then
        echo ISSAC FILE
fi
DIST=`saclhdr -DIST $1`
echo $DIST
```

The command is run directly from the command line as:

`saclhdr [-?] [-h] -Cmd[s] -NL SAC_ALPHA_OR_BINARY_FILE`

The -Cmd is one of the SAC header items listed in the description of **`shwsac`**. The SAC file can either be machine dependent binary or in the SAC alpha format. The program attempts to determine the data type.

In addition to the SAC header values, we recently added the *NZMON* and *NZDAY* to permit output of month and day.

If the **`-NL`** flag is not used, then only the first command is output using a C printf statment, such as *printf("%f",fval)* - no newline is placed in the output, which permits it use as a SHELL variable.

If the **`-NL`** flat is invoked, more than one command can be evaluated, and a space is output in between the fields using a C printf statement, such as *printf("%f ",fval)*, with the entire stream terminated by a newline.

The output is free format. If the SAC header is set to the default no-value or *-12345.*, *-12345* or *"-12345"* for real, integer or string values, then these appear on the output.

Special formatting is used for the *NZYEAR*, *NZJDAY*, *NZMON*, *NZDAY NZHOUR*, *NZMIN* and *NZMSEC* fields to ensure that the complete field widths of 4, 3, 2, 2, 2, 2, and 3, respectively are used. Thus successive queries can lead to the time stamp:

2001 002 05 10 03 010

This is useful for defining unique trace names or wavefrom directories directly from the header:

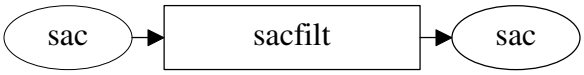
```
#!/bin/sh

#####
#               query the SAC header to define a name for storing the receiver function
#               for this station
#####
ALP=1.0
for STA in *BHZ*
do
                BASE=`basename ${STA} .BHZ`
                saciterd -ALP ${ALP} -P 0.10 -N ${BASE}.BHR -D ${BASE}.BHZ -D 10
                KSTNM=`saclhdr -KSTNM ${STA}`
                NZYEAR=`saclhdr -NZYEAR ${STA}`
                NZJDAY=`saclhdr -NZJDAY ${STA}`
                NZHOUR=`saclhdr -NZHOUR ${STA}`
#                be careful here in that the trace time may not be the origin time
#                does the directory for this stations' receiver functions exist
                if [ -d ../${STA}RFTN ]
                then
                echo exists
                else
                mkdir ../${STA}RFTN
                fi
#####
#               rename the decon.out file and move it to the station directory
#####
                mv decon.out ../${STA}RFTN/${NZYEAR}${NZJDAY}${NZHOUR}${STA}${ALP}
done
```

The *-NL* flag was introduced to permit an quick evaluation of the contents of the SAC header without a lot of SHELL programming. This example examines the header values of some receiver functions.

```
#!/bin/sh
for i in */*.1.0
do
                saclhdr -NL -KSTNM -KCMPNM -USER0 -USER4 -USER6 $i
done
```

11. sacfilt

This program applies or removes a general instrument/filter response define in terms of poles and zeros. For compatibility with routines that convert SEED or GSE3.0 to SAC, the pole-zero response is defined in SAC format.



Program control is through command line flags:

sacfilt *[flags]*, where the command flags are
-DEMEAN (default false) remove mean before filter
-TAPER (default false) apply taper before filter
-FREQLIMITS f1 f2 f3 f4 apply a taper for deconvolution (-R) This cubic taper ensures that the deconvoled spectrum passband is [f2,f3] and that the spectum is zero for **f < f1** and **f > f4** with a cubic taper int the region [f1,f2] and [f3,f4].
-A (default true) apply filter
-R (default false) remove filter
-PZ pole_zero_file (none) SAC response file
-SACIN *binary_sac_input* file (none)
-SACOUT *binary_sac_output* file (none)
-h
-? Online help concerning program usage

In addition the program sets several of the SAC header values in the file *binary_sac.out* if the instrument response is removed:

USER1 - minimum period in the passband
USER2 - maximum period in the passband
KUSER1 - set to the string *PER_MIN*.
KUSER2 - set to the string *PER_MAX*.
These fields are set when removing the instrument response (**-R**) so that other programs, such as **sacmft96** and **sacpom96**, only use frequencies within the passband of deconvolution. The use of frequency limits ensures a stable deconvolution by a zero-phase

bandbass.

12. saccvt

This program addresses the problem of transporting SAC binary files between a SPARC, or other machine using IEEE big-endian INTEL little-endian representations of numbers. This utility thus performs the necessary byte swaps to accomplish this.

Program control is through command line flags:

saccvt *[flags]*, where the command flags are
-I (default none) intelligently guess whether to convert
-h
-? Online help concerning program usage

The use of the program is illustrated by executing **saccvt -h** or **saccvt -?**, which gives:

```
 Convert SAC binary  IEEE to INTEL format
 Convert SAC binary  INTEL to IEEE format
  All 4 byte integers and floats (a,b,c,d) are
  transposed to (d,c,b,a)
Example: saccvt < SAC_BINARY > tmp ; mv tmp SAC_BINARY
-I   (default none)  intelligently guess whether to convert
-h   (default none)  this help message
-?   (default none)  this help message
```

Ultimately this program will be modified to permit the files to be converted to be given on the command line. The reason for the two stage process outlined above is to ensure that the conversion is a conscious act. The other alternative is to use the **sactosac** and **asctosac** routines to convert the native binary SAC file to ASCII, transfer the ASCII between the machines, and then convert the ASCII to the other machine's binary SAC file format. The use of **saccvt** is the preferred mechanism.

The **-I** flag is the latest addition to the code. If this flag is invoked, then the file is examined by looking for the pattern -12345. in the floating point header values or the integer -12345 in the integer header values. if this is NOT seen then the file is converted. Otherwise if is not. I usually run this program in a shell script:

```
for i in *sac
do
        saccvt -I < i > tmp; mv tmp i
done
```

This will ensure that all SAC files in the current directory are in the format for the local architecture.

CHAPTER 8
EARTH MODEL FILES

1. Introduction

This chapter describes the earth model format in detail. Preliminary versions of programs that use the earth model format are documented.

2. model96

The **model96** format is described in Chapter 1 for flat-layered media. Chapter 5 discusses the extension to 2-D varying interfaces and vertically varying velocity functions in the context of asymptotic ray tracing. The focus of this chapter is on layered media with constant velocity layers.

3. shwmod96

This program plots an earth model in the **model96** format.

The program is invoked as

shwmod96 *[flags] model96_file[s]*
- **-XLEN** *xlen* (default 6.0) Length of horizontal axis
- **-YLEN** *ylen* (default 6.0) Length of depth axis
- **-X0** *x0* (default 2.0) (x0,y0) are lower left corner
- **-Y0** *y0* (default 1.0)
- **-K** *kolor* (default 1) Profile in color. If *kolor* < *0* plot models with progression from red to blue.
- **-VMIN** *vmin* (default 2.0) Minimum value of horizontal
- **-VMAX** *vmax* (default 5.0) Minimum value of horizontal
- **-ZMIN** *zmin* (default 0.0) Minimum value of depth axis
- **-ZMAX** *zmax* (default 60.) Minimum value of horizontal axis
- **-W** *width* (default 0.001") Line width in inches
- **-NOLABX (default label X) Do not label X-axis'**
- **-NOLABY (default label Y) Do not label Y-axis'**
- **-P** (default S) plot P-velocity

-S (default S) plot S-velocity
-D (default S) plot density
-? (default none) this help message
-h Online help.

NOTE: WORKS ONLY FOR ISOTROPIC EARTH MODELS

4. timmod96

This program predicts and plots the first arrival times of the P, SV and SH arrivals for a given **model96** file[s].

The program is invoked as

timmod96 *[flags] model96_file[s]*
-XLEN *xlen* (default 6.0) Length X-axis
-YLEN *ylen* (default 6.0) Length Y-axis
-VRED *vred* (default not used) reduction velocity
-HS *hs* (default 0.0) Source depth km
-HR *hr* (default 0.0) Receiver depth km
-TMIN *tmin* (default 0.0) Minimum value of time
-TMAX *tmax* (default 100.0) Maximum value of time
-XMIN *xmin* (default 0.0) Minimum value of distance
-XMAX *xmax* (default 300.0) Maximum value of distance
-X0 *x0* (default 2.0) x-position of lower left corner
-Y0 *y0* (default 1.0) y-position of lower left corner
-K *kolor* (default 1) Profile in color. If *kolor < 0* plot models with progression from red to blue
-K kolor (default 1) Profile in color
if kolor < 0 use red->blue progression
-P (default) Plot P times
-SV (default P) Plot SV times
-SH (default P) Plot SH times
-W *width* (default 0.001") Line width in inches
-?
-h Online help.

NOTE: WORKS ONLY FOR ISOTROPIC FLAT EARTH MODELS

The program also creates an output file *TIMMOD96.TXT* which has four columns of output: distance (km), P time (sec), SV time (sec) and SH time (sec). Of these only one set of travel time-distance values are plotted according to the use of the **-P**, **-SV** and **-SH** flags. The **-VRED** *vred* can change the plot to a reduced travel time plot, but this does not affect the listing of times in the file *TIMMOD96.TXT*.

The graphics file *TTIMMOD96.PLT* is also created.

Note also that this is preliminary code and that the presentation must be cleaned up.

5. mkmod96

This program currently interactively creates a **`model96`** file for a 1-D velocity model with constant velocity layers. One use of the program is to prototype the correct model format and then to use an *editor* to add detail to the model.

The following examples indicates the use of the program. User input is indicated by the **`Courier Bold`** font and program output by the `Courier` font.

```
mkmod96
 Write creating model96 file for isotropic constant velocity layers, 1-D model
 Enter name of the earth model file
model1.d
  Model file is :model1.d
 Enter model comment
Isotropic model
  Comment is    :Isotropic model
 Enter 0 for flat earth model
       1 for spherical earth model
0
  Model flat/sph: 0
 Enter Velocity Model, EOF to end:
  H     VP     VS      RHO    QP QS ETAP ETAS FREFP FREFS
 (km) (km/s) (km/s) (gm/cm^3) -- -- ---- ----  (Hz)  (Hz)
0     7     4.5      2.7    100 100 0.0  0.0  1.0 1.0
  40.  7.  4.5  2.70000005  100.  100.  0.  0.  1.  1.
 0     9     5.7      3.3    1000 1000 0.0 0.0 1.0 1.0
 0.  9.  5.69999981  3.29999995  1000.  1000.  0.  0.  1.  1.
(enter a End Of File - a CTRL Z RETURN  in DOS or usually a CTRL D on UNIX/LINUX
 Overwriting Existing model File


mkmod96
 Write creating model96 file for isotropic constant velocity layers, 1-D model
 Enter name of the earth model file
model2.d
  Model file is :model2.d
 Enter model comment
Spherical isotropic model
  Comment is    :Spherical isotropic model
 Enter 0 for flat earth model
       1 for spherical earth model
1
  Model flat/sph: 1
 Enter Velocity Model, EOF to end:
  H     VP     VS      RHO    QP QS ETAP ETAS FREFP FREFS
 (km) (km/s) (km/s) (gm/cm^3) -- -- ---- ----  (Hz)  (Hz)
-50   0.30    0.0     0.001  0   0   0    0   1.0  1.0
 -50.  0.300000012  0.  0.00100000005  0.  0.  0.  0.  1.  1.
-10   0.31    0.0     0.001  0   0   0    0   1.0  1.0
 -10.  0.310000002  0.  0.00100000005  0.  0.  0.  0.  1.  1.
40     6     3.5      2.7    100 100 0.0  0.0  1.0 1.0
  40.  6.  3.5  2.70000005  100.  100.  0.  0.  1.  1.
 0     8     4.7      3.3    1000 1000 0.0 0.0 1.0 1.0
  0.  8.  4.69999981  3.29999995  1000.  1000.  0.  0.  1.  1.
(enter a End Of File - a CTRL Z RETURN  in DOS or usually a CTRL D on UNIX/LINUX
 Overwriting Existing model File
```

Note that the second model file used the negative layer thicknesses of the first two layers to indicate that the model extended above the reference datum - the surface of the earth. To plot the velocity profiles **shwmod96** is used. Note that **shwmod96** correctly plots the model with respect to the surface datum.

```
shwmod96 -P -VMIN 0 -VMAX 10 -ZMIN -100 -ZMAX 50 -K 2 model*.d
```

The `model*.d` is a UNIX/LINUX wild-card expansion. This will also work under DOS because of the use of the gcc/g77 compilers.

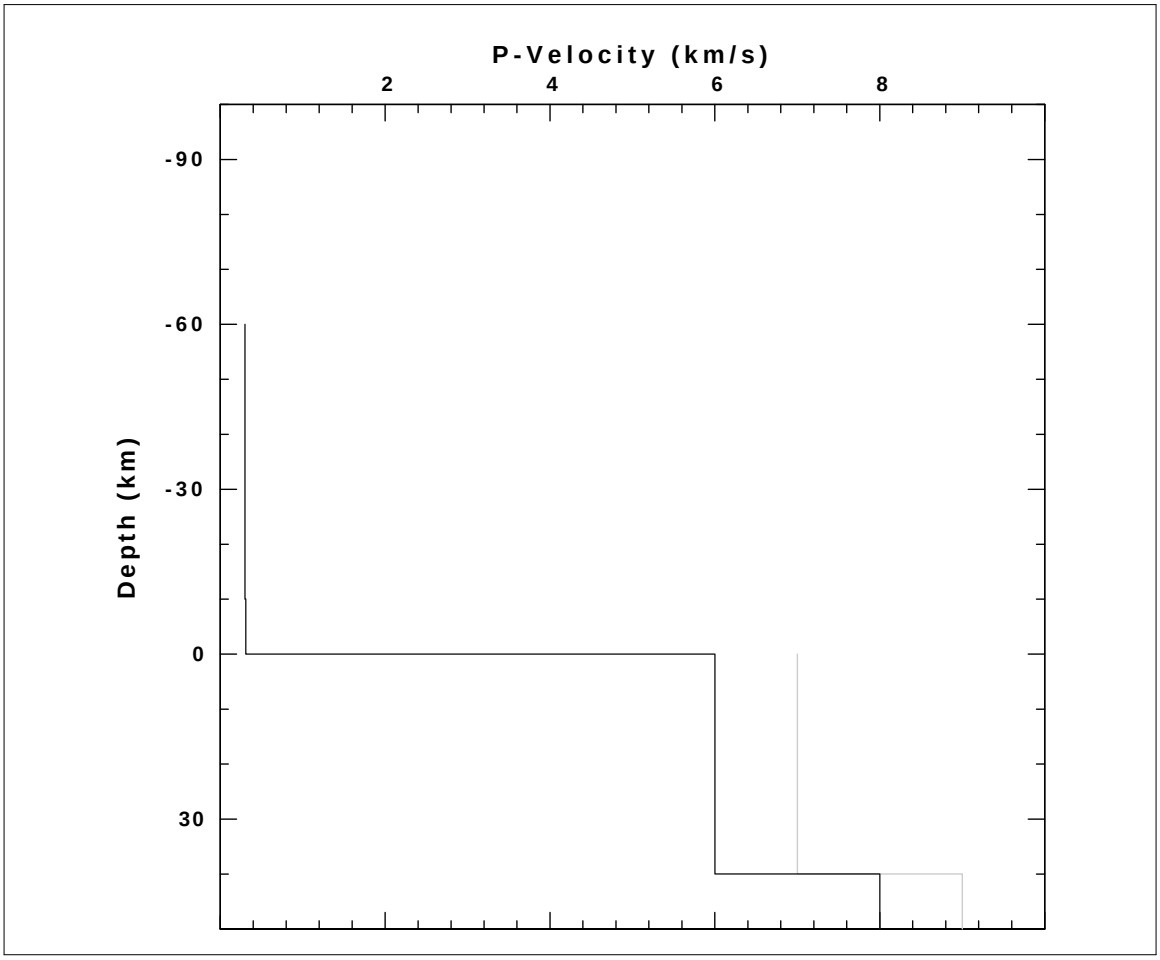


Fig. 1. Plot of the models files produced using **mkmod96**.

6. time96

This program predicts the first arrival times of the P, SV and SH arrivals for a given **model96** file[s]. This program is designed to used in the manner of **udfdd** in that output can be placed in a SHELL variable.

As an example, assume that you wish to make synthetics that start 100 seconds before the predicted P-wave first arrival. Also assume that you wish to place the P travel time, relative to the origin time in to the SAC header value *A* and that you wish do define a parameter *T0* for a synthetic seismogram program. This can be done as following, in this BASH SHELL fragment:

```
for GCARC in \
20 30 40
do
A="time96 -M AK135sph.mod -GCARC ${GCARC} -T -EVDP ${HS}"
T0="echo $A | awk '{printf "%f", $1 - 100}' "
done
```

The program is invoked as

time96 *[flags]*
-GCARC *gcarc* (default none) epicentral dist deg
-DIST *dist* (default none) epicentral dist km
-EVDP *evdp* (default 10.0) source depth km
-T (default true) output travel time
-RAYP (default false) output ray param
-GEOM (default false) output geometrical spreading
-TS (default false) output ray T*
-P (default true) output for P-wave
-SV (default false) output for SV-wave
-SH (default false) output for SH-wave
-pP (default false) output for teleseismic pP
-sP (default false) output for teleseismic sP
-sS (default false) output for teleseismic sS
-M *model_name* Model96 velocity model
-h this command help
-? this command help

An earth flattening approximation is used for spherical models.

Ultimately this program could be used to provide the geometrical spreading for a spherical Earth model.

7. refmod96

This program predicts and plots the first arrival times of the P, SV and SH arrivals, as well as reflection times, for a given **model96** file[s].

This special purpose program is mean to be used with the *plotrecordsection* or *refr* command plot output of **gsac**.

The program is invoked as

refmod96 *[flags]*
-XLEN *xlen* (default 6.0) Length X-axis
-YLEN *ylen* (default 6.0) Length Y-axis
-VRED *vred* (default not used) reduction velocity
-TMIN *tmin* (default 0.0) Minimum value of time
-TMAX *tmax* (default 100.0) Maximum value of time
-XMIN *xmin* (default 0.0) Minimum value of distance
-XMAX *xmax* (default 300.0) Maximum value of distance
-X0 *x0* (default 2.0) x-position of lower left corner
-Y0 *y0* (default 1.0) y-position of lower left corner
-W *width* (default 0.001) Width of line (inch) for model plot
-KR *kr* (default 4) Color of reflection
-KF *kf* (default 2) Color of first arrival

-M model (required) model96 file name
-P (default) Plot P times
-SV (default P) Plot SV times
-SH (default P) Plot SH times
-NOBOX (default none) do not plot bounding frame
-NMULT *nmult* (default 1) number of reflection multiples
-LEG (default none) Put in file legend
-HS hs (default 0.0) source depth in km
-? (default none) this help message
-h (default none) this help message

NOTE: WORKS ONLY FOR ISOTROPIC FLAT EARTH MODELS

8. prfmod96

This program contour plots a 2-D S-wave velocity model using 1-D **model96** velocity models specified as a function of latitude and longitude.

The program is invoked as

prfmod96 *[flags]*
-XLEN *xlen* (default 6.0) Length X-axis
-YLEN *ylen* (default 6.0) Length Y-axis
-X0 x0 (default 2.0) x-position of lower left corner
-Y0 y0 (default 1.0) y-position of lower left corner
-ZMAX *zmax* (default 100.0) maximum depth (km) for plot
-LAT1 *lat1* (default none) left coordinates of plot
-LON1 *lon1* (default none)
-LAT2 *lat2* (default none) left coordinates of plot
-LON2 *lon2* (default none)
-W *width* (default 50) Width (km) of projection
-C *control* (required) control name
-KV *control_value_file* If not defined the contour VALUES
are 2.5 - 6.0 in increments of 0.25 km/sec.
The file has one value per line in increasing order
-KA *control_label_file* If not defined the contour LABELS
are 2.5 - 6.0 in increments of 0.50 km/sec
The file has one value per line in increasing order
lat lon model96_file
-? (default none) this help message
-h (default none) this help message

NOTE: WORKS ONLY FOR ISOTROPIC FLAT EARTH MODELS

An example of the use of this program in in the following BASH SHELL script. This script creates the files for the annotation of the plot. The data set consists of 1-D models for Korea and creates a profile by using 1-D models within ±50 km of the selected profile.

The contents of the file *control* which associates latitude, longitude, station name and velocity model is given here. Note that FORTRAN requires the use of single quotes to delimit the station and file names in the script. The file *WW* contains entries such as:

```
36.569 128.706 'AND' '../FINV/FinvAND/end.mod'
37.968 124.630 'BRD' '../FINV/FinvBRD/end.mod'
35.249 129.113 'BUS' '../FINV/FinvBUS/end.mod'
36.823 127.257 'CEA' '../FINV/FinvCEA/end.mod'
36.773 127.119 'CHA' '../FINV/FinvCHA/end.mod'
```

The processing script, *NDOIT*, is

```
#!/bin/sh
#####
#           define plot limits
#####
VMIN=2.5
VMAX=5.5
ZMAX=40.0
WIDTH=50.0
#####
#           define the contour values
#####
cat > kv_file << EOF
3.00
3.25
3.50
3.75
4.00
4.25
4.50
EOF
#####
#           define the contour values to be annotated. To have a clean
#           appearance, only a few contour labels are plotted
#####
cat > ka_file << EOF
3.0
3.5
4.0
4.5
EOF

prfmod96 -YLEN 3 -ZMAX ${ZMAX}  -C WW ${LL} -VMIN ${VMIN} \
           -VMAX ${VMAX} -W ${WIDTH} -KA ca_file -KV cv_file
cat PRFMOD96.PLT | plotnps -F7 -W10 -EPS -K >  PROF.eps
```

The resulting plot is shown in Figure 2. Note that the plotting routine is not perfect in that station names can overlap. This can be adjusted using Adobe illustrator.

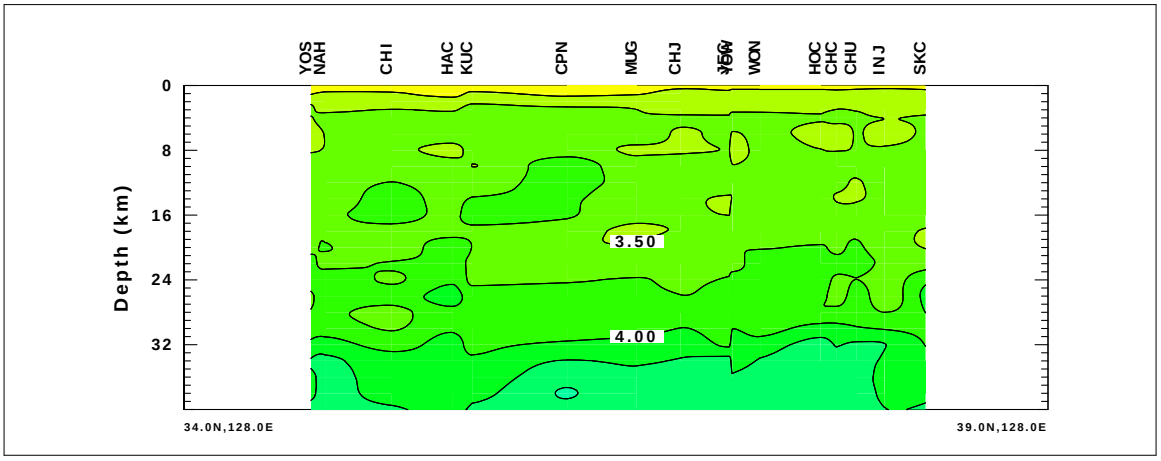


Fig. 2. Plot of the models files produced using **`mkmod96`**.

CHAPTER 9
TRANSVERSE ISOTROPY REVISED

1. Introduction

This chapter describes the implementation of synthetic seismogram codes for transversely isotropic media. Eventually the documentation will be included in the chapters corresponding to the particular technique. The wavenumber integration codes for Chanpter 3 and the surface wave cords for Chapter4 have been adapted for use in TI media, with the name changes indicted in the following table:

tspec96	Extension to hspec96
tspec96p	Extension to hspec96p
trftn96	Extension of hrftn96
ti2ismod	Convert TI model to best Isotropic Model
is2timod	Convert Isotropic model to TI Model
tmkmod96	Similar to mkmod96 - interactively create TI model
ttimmod96	Extension to timmod96
ttime96	Extension to time96
tshwmod96	Extension to shwmod96
tfmech96	Extension to fmech96
tprep96	TI version of sprep96
tdisp96	TI version of sdisp96
tregn96	TI version of sregn96
tlegn96	TI version of slegn96
tpulse96	TI version of spulse96
tdpsrf96	TI version of sdpsrf96
tdpegn96	TI version of sdpegn96
tdpder96	TI version of sdpder96
tcomb96	TI version of scomb96

The term *Extension* means that the program works with both transverse isotropic (TI) and isotropic models. Computation times for an isotropic model are significantly faster when using the corresponding purely isotropic implementation. Certain programs have been modified to use either a TI or isotropic model (e.g., **`hprep96`**, **`hspec96`**)

2. TI model

A transversely isotropic velocity model with a vertical axis of symmetry is defined by five elastic constants (Love, 1944). In the Love notation, the moduli of elasticity are A, C, F, L and N. I define the velocities $VPV = \sqrt{C/\rho}$, $VPH = \sqrt{A/\rho}$, $VSV = \sqrt{L/\rho}$, $VSH = \sqrt{N/\rho}$ and $VPF = \sqrt{F/\rho}$. Four of these have simple meanings. VPV is the velocity of a vertically propagating quasi-P wave, and VSV is the velocity of a vertically propagating quasi-S wave for which there is no distinction between quasi-SV and quasi-SH. VPH is the velocity of a horizontally propagating quasi-P wave. VSH is the velocity of the horizontally propagating quasi-SH wave. For reference, the quasi-SV wave propagating horizontally has the velocity VSV. The VPF is a constructed velocity related to the F modulus.

Since an isotropic medium is just a special case of a TI model, the TI moduli are related to the Lamé parameters by the simple relations $A = C = \lambda + 2\mu$, $F = \lambda$ and $L = N = \mu$.

Because a TI medium requires five elastic constants for a complete description, incorporation of anelasticity should also require five different Q values. In the wavenumber integration and surface-wave dispersion programs, causal Q is introduced. *A design decision was made to describe the* Q*-effect in terms of only two values - Q_P and Q_S,* which are used in the isotropic mdeia. Since anisotropy in the Earth exists in the upper mantle as a subtle perturbation to the isotropic earth, the decision to have only a Q_P and a Q_S is justifiable on the basis of lack of knowledge of actual TI-Q in the Earth and because any effect may be subtle. Before introducing a design decision for implementing Q in a TI model, a review of some concepts is presented.

Anderson and Archambeau (?, 1966X) discussed the attenuation of P- and S-waves in terms of the bulk modulus κ and rigidity μ. At a fixed frequency, the complex V_P and V_S are defined in terms of complex bulk and shear moduli, $\kappa + i\kappa^*$ and $\mu + i\mu^*$:

$$V_P = \sqrt{\frac{\kappa + \frac{4}{3}\mu + i(\kappa^* + \frac{4}{3}\mu^*)}{\rho}}$$

$$V_S = \sqrt{\frac{\mu + i\mu^*}{\rho}}$$

Assuming that $\kappa^* \ll \kappa$ and $\mu^* \ll \mu$, then

$$V_P \approx \sqrt{\frac{\kappa + \frac{4}{3}\mu}{\rho}}\left(1 + \frac{i}{2}\,\frac{\kappa^* + \frac{4}{3}\mu^*}{\kappa + \frac{4}{3}\mu}\right)$$

$$\equiv V_P(\text{elastic})\left(1 + \frac{i}{2Q_P}\right)$$

and

$$V_S \approx \sqrt{\frac{\mu}{\rho}}\left(1 + \frac{i}{2}\,\frac{\mu^*}{\mu}\right)$$

$$\equiv V_S(\text{elastic})\left(1+\frac{i}{2Q_S}\right)$$

which leads to the definitions

$$\frac{1}{Q_P}\equiv\frac{\kappa^*+\frac{4}{3}\mu^*}{\kappa+\frac{4}{3}\mu}$$

and

$$\frac{1}{Q_S}\equiv\frac{\mu^*}{\mu}$$

Anderson and Archambeau (196?, ?) note that if energy loss is due only to shear, $\kappa^* = 0$, and that as a consequence

$$\frac{Q_P}{Q_S}=\frac{3}{4}\frac{V_P^2(\text{elastic})}{V_S^2(\text{elastic})}$$

When $\lambda = \mu$, then $Q_P / Q_S = 2.25$.

To implement attenuation into the TI model, three design decisions are made:

- $\frac{A^*}{A}=\frac{C^*}{C}=\frac{1}{Q_P}$
- $\frac{L^*}{L}=\frac{N^*}{N}=\frac{1}{Q_S}$
- F * is chosen in a manner to reduce to the isotropic λ * so that the model synthetics reduce to isotropic synthetics as the TI model $\rightarrow$ isotropic.

To accomplish this we start with equations (8.191) and (8.192) of Dahlen and Tromp (1998) that relate the TI constants to the best isotropic model bulk moduli:

$$\kappa=\lambda+\frac{2}{3}\mu=\frac{1}{9}(C+4A-4N+4F)$$

$$\mu=\frac{1}{15}(C+A+6L+5N-2F)$$

Following Archambeau and Anderson (196?, ?), at a fixed frequency,

$$\kappa^*=\lambda^*+\frac{2}{3}\mu^*=\frac{1}{9}(C^*+4A^*-4N^*+4F^*)$$

$$\mu^*=\frac{1}{15}(C^*+A^*+6L^*+5N^*-2F^*)$$

Noting that a causal Q operator also makes the real part of the complex velocity frequency dependent, two factors, $f_a(\omega,\omega_{ref},Q_P^{-1})$ and $f_s(\omega,\omega_{ref},Q_S^{-1})$ are introduced that convert the frequency independent purely elastic moduli to the frequency dependent complex moduli. The factors $\rightarrow 1$ as $Q\rightarrow\infty$. The complex F is obtained from the definition

$$2F\equiv Cf_a^2+Af_a^2+6Lf_b^2+5Nf_b^2-15\mu f_b^2$$

where **μ**, **A**, **C**, **L** and **N** are real constants. Algebraic substitution shows that in the isotropic limit for the special case of $\kappa^* = 0$ $\lambda^* / \lambda = Q_P^{-1}$ and $\mu^* / \mu = Q_S^{-1}$. This approach is taken since the P- and Q- causal atenuation operators may have different reference frequencies.

3. tmkmod96

This program interactively creates a **model96** file for a 1-D velocity model with constant velocity layers. One use of the program is to prototype the correct model format and then to use an *editor* to add detail to the model.

The following examples indicates the use of the program. User input is indicated by the **Courier Bold** font and program output by the Courier font.

```
tmkmod96
 Write creating model96 file for transverse isotropic
 constant velocity layers, 1-D model
 Enter name of the earth model file
MVDH.mod.ti
  Model file is :MVDH.mod.ti
 Enter model comment
 Modified Van der Heiden beryl TI model - P vel scaled to 6 km s^-1
  Comment is    :
  Modified Van der Heiden beryl TI model - P vel scaled to 6 km s^-1
 Enter 0 for flat earth model
       1 for spherical earth model
0
  Model flat/sph: 0
 Enter Velocity Model, EOF to end:
  H     VPV    VSV     RHO    QP QS ETAP ETAS FREFP FREFS
 (km) (km/s) (km/s) (gm/cm^3) -- -- ---- ----  (Hz)  (Hz)
        VPH    VSH     VPF
      (km/s) (km/s)  (km/s)
1.0000   5.7530   3.0220   2.7000   0.00     0.00     0.00     0.00     1.00     1.00
6.1430   3.4730   3.0430
  1.  5.75299978  3.02200007  2.70000005  0.  0.  0.  0.  1.  1.
  6.14300013  3.47300005  3.04299998
 Creating the model file:MVDH.mod.ti
 Overwriting Existing model File
```

The model file created, MVHD.mod.ti is the following:

```
MODEL.01
 Modified Van der Heiden beryl TI model - P vel scaled to 6 km s^-1
TRANSVERSE ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM)   VPV(KM/S)  VSV(KM/S) RHO(GM/CC)  QP    QS      ETAP     ETAS    FREFP    FREFS
          VPH(KM/S)  VSH(KM/S)  VPF(KM/S)
 1.0000    5.7530   3.0220   2.7000   0.00     0.00     0.00     0.00     1.00     1.00
           6.1430   3.4730   3.0430
```

4. is2timod

This program reads an isotropic model and converts it into a TI model. One use of this program is to do the hard work of converting a model that is mostly isotropic to one that is in the TI model format. For example, one may wish an upper mantle model that only has true transverse isotropy within a narrow region of the upper mantle. The TI model file can then be edited to give the correct TI model values to selected layers.

This program does not have an on-line help with the **-h** or **-?** command line flags. The program is just a filter.

Consider the following isotropic model contained in the filter **model.is**.

```
MODEL
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
HR      VP       VS  RHO QP  QS  ETAP ETAS FREFP FREFS
40  5.8383  3.3408 2.7000 0.00 0.00 0.00 0.00 1.00 1.00
 0  8       4.7    3.3    0.0  0.0  0.0  0.0  1.0   1.0
```

Running the command

```
is2timod < model.is > model.ti
```

creates the equivalent model file in TI format, **model.ti**:

```
MODEL.01
TEST MODEL
TRANSVERSE ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM)   VPV(KM/S)  VSV(KM/S) RHO(GM/CC)  QP  QS     ETAP    ETAS   FREFP   FREFS
          VPH(KM/S)  VSH(KM/S)  VPF(KM/S)

40.0000    5.8383   3.3408   2.7000   0.00    0.00    0.00    0.00    1.00    1.00
           5.8383   3.3408   3.4298
 0.0000    8.0000   4.7000   3.3000   0.00    0.00    0.00    0.00    1.00    1.00
           8.0000   4.7000   4.4520
```

5. ti2ismod

This program reads a transverse isotropic model and converts it into an equivalent isotropic model. This conversion uses equations (8.191) and (8.192) of Dahlen and Tromp (1998):

$$\kappa = \lambda + \frac{2}{3}\mu = \frac{1}{9}(C + 4A - 4N + 4F)$$

$$\mu = \frac{1}{15}(C + A + 6L + 5N - 2F)$$

The use of this program may be to create an equivalent isotropic model so that isotropic synthetic seismograms can be compared to the TI ones.

This program does not have an on-line help with the **-h** or **-?** command line flags. The program is just a filter.

Consider the following transversely isotropic model contained in the file **model.TI**:

```
MODEL.01
TEST MODEL
TRANSVERSE ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
   H(KM) VPV(KM/S) VSV(KM/S) RHO(GM/CC) QP   QS  ETAP  ETAS  FREFP  FREFS
         VPH(KM/S) VSH(KM/S) VPF(KM/S)
 40.0000   5.7530   3.0220   2.7000  0.00  0.00  0.00  0.00   1.00   1.00
           6.1430   3.4730   3.0430
  0.0000   8.0000   4.7000   3.3000    0.     0. 0.00  0.00   1.00   1.00
           8.0000   4.7000   4.4520
```

Running the command

`ti2ismod < model.TI > model.IS`

creates the following best isotropic model file, **`model.IS`** :

```
MODEL.01
TEST MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM) VP(KM/S) VS(KM/S) RHO(GM/CC)  QP   QS  ETAP  ETAS FREFP  FREFS
40.0000   5.8383  3.3408   2.7000   0.00 0.00  0.00  0.00  1.00   1.00
 0.0000   8.0000  4.7000   3.3000   0.00 0.00  0.00  0.00  1.00   1.00
```

6. tshwmod96

This program plots an earth model in the **`model96`** format either for the isotropic or transverse isotropic 1-D constant velocity layered specification.

The program differs from **`shwmod96`** in that the plotted depth values increase downward instead of decreasing. If the model is transverse isotropic the VPH and VSH velocities, corresponding to a horizontally propagating quasi-P and quasi-SH, respectively, are plotted with dashed lines.The VPV and VSV velocities, corresponding to vertically propagating quasi-P and quasi-Sv or quasi-SH, respectively, as well as the P- and S-wave velocities for an isotropic medium are plotted with solid line segments.

Although the program plots VPv, VSV, VPH and VSH the pseudo-velocity VPF is not plotted. Rather the parameter $\eta = \mathbf{F/A} - \mathbf{2L}$ (Dahlen and Tromp, 1999) is plotted using

the command line option −E. The parameter $\eta = 1$ for an isotropic medium.

The output is a CALPLOT plot file with name SHWMOD96.PLT.

The program is invoked as

tshwmod96 *[flags] model96_file[s]*
-XLEN *xlen* (default 6.0) Length of horizontal axis
-YLEN *ylen* (default 6.0) Length of depth axis
-X0 *x0* (default 2.0) (x0,y0) are lower left corner
-Y0 *y0* (default 1.0)
-K *kolor* (default 1) Profile in color. If *kolor* < *0* plot models with progression from red to blue.
-VMIN *vmin* (default 2.0) Minimum value of horizontal
-VMAX *vmax* (default 5.0) Minimum value of horizontal
-ZMIN *zmin* (default 0.0) Minimum value of depth axis
-ZMAX *zmax* (default 60.) Minimum value of horizontal axis
-W *width* (default 0.001") Line width in inches
-NOLABX (default label X) Do not label X-axis'
-NOLABY (default label Y) Do not label Y-axis'
-P (default S) plot P-velocity
-S (default S) plot S-velocity
-D (default S) plot density
-E (default S) plot eta
-? (default none) this help message
-h Online help.

The relation between seismic moment, M_0, and moment magnitude, M_W used is

$$\log_{10} M_0 = 1.5 M_W + 16.05$$

The following shell script plots the TI and equivalent isotropic models in two displays. the first plots the velocities, densities and eta side-by-side to produce the plot file TIMOD1.PLT with different colors representing the different models. The other overlays all model parameters in the file model.TI onto a single frame with different colors representing the two velocities, density and η.

```
#!/bin/sh

#####
#               Define colors
#####
BLACK=1
RED=2
GREEN=3
BLUE=4
RAINBOW=-1

#####
#                P S density Eta in side-by-side frames
#####
tshwmod96 -X0 1.0 -XLEN 2.5  -VMIN 5.0 -VMAX 9.0 -ZMIN 0  -ZMAX 50. -P \
        -K ${RAINBOW}  model.TI model.ti
mv SHWMOD96.PLT mp
tshwmod96 -X0 3.5 -XLEN 2.5  -VMIN 2.0 -VMAX 6.0 -ZMIN 0  -ZMAX 50. -S \
        -K ${RAINBOW} model.TI model.ti
reframe -N1 -O -XL3500 < SHWMOD96.PLT > ms
tshwmod96 -X0 6.0 -XLEN 2.5  -VMIN 2.0 -VMAX 4.0 -ZMIN 0  -ZMAX 50. -D \
        -K ${RAINBOW} model.TI model.ti
reframe -N1 -O -XL6000 < SHWMOD96.PLT > md
tshwmod96 -X0 8.5 -XLEN 2.5  -VMIN 0.0 -VMAX 1.5 -ZMIN 0  -ZMAX 50. -E \
        -K ${RAINBOW} model.TI model.ti
reframe -N1 -O -XL8500 < SHWMOD96.PLT > me
cat mp ms md me > TIMOD1.PLT
rm -f mp ms me md
rm -fr SHWMOD96.PLT
plotnps -EPS -K < TIMOD1.PLT > timod1.eps

#####
#               Plot P S Density and Eta on one frame - use different colors for each
#               cut off the top
#####
tshwmod96 -X0 1.0 -XLEN 4.0  -VMIN 0.0 -VMAX 9.0 -ZMIN 0  -ZMAX 50. -P -K \
        ${BLACK} model.TI
cat SHWMOD96.PLT >  mm
tshwmod96 -X0 1.0 -XLEN 4.0  -VMIN 0.0 -VMAX 9.0 -ZMIN 0  -ZMAX 50. -S -K \
        ${RED} model.TI
cat SHWMOD96.PLT >> mm
tshwmod96 -X0 1.0 -XLEN 4.0  -VMIN 0.0 -VMAX 9.0 -ZMIN 0  -ZMAX 50. -D -K \
        ${GREEN} model.TI
cat SHWMOD96.PLT >> mm
tshwmod96 -X0 1.0 -XLEN 4.0  -VMIN 0.0 -VMAX 9.0 -ZMIN 0  -ZMAX 50. -E -K \
        ${RED} model.TI
cat SHWMOD96.PLT >> mm
reframe -YH7200 -O <  mm > TIMOD2.PLT
rm mm
rm SHWMOD96.PLT
plotnps -EPS -K < TIMOD2.PLT > timod2.eps
```

The resultant plots are

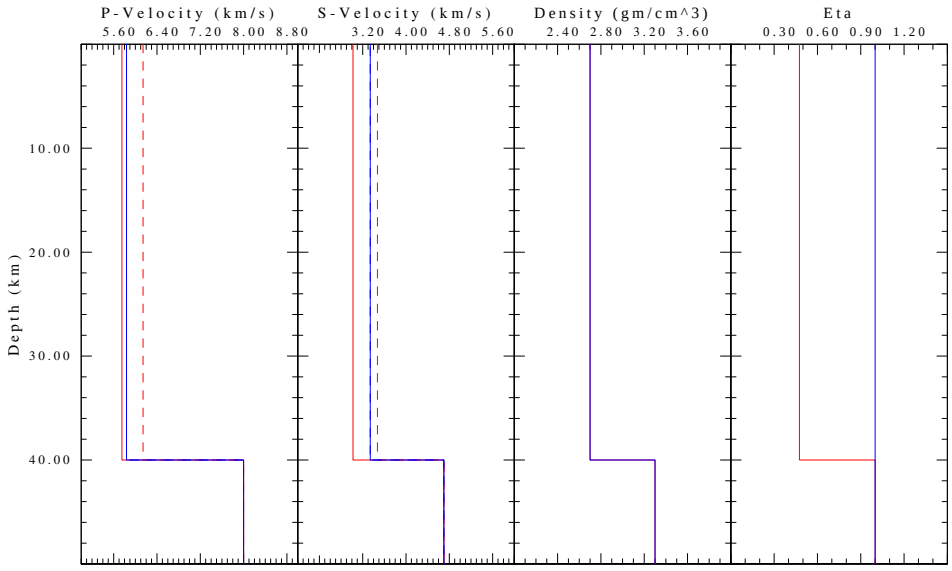


Fig. 1. TIMOD1.PLT

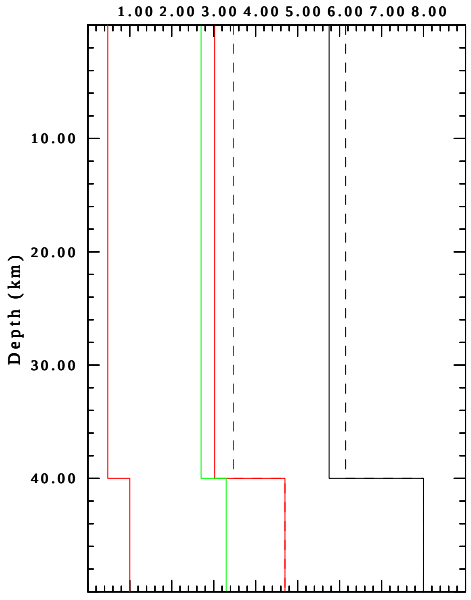


Fig. 2. TIMOD2.PLT

7. ttime96

This program predicts the first arrival times of the P, SV and SH arrivals for a given **`model96`** file[s]. This program is designed to used in the manner of **`udfdd`** in that output can be placed in a SHELL variable.

As an example, assume that you wish to make synthetics that start 100 seconds before the predicted P-wave first arrival. Also assume that you wish to place the P travel time,

relative to the origin time in to the SAC header value *A* and that you wish do define a parameter *T0* for a synthetic seismogram program. This can be done as following, in this BASH SHELL fragment:

```
for GCARC in \
20 30 40
do
A="ttime96 -M tak135sph.mod -GCARC ${GCARC} -T -EVDP ${HS}"
T0="echo $A | awk '{printf "%f", $1 - 100}' "
done
```

The program is invoked as

ttime96 *[flags]*
- **-GCARC** *gcarc* (default none) epicentral dist deg
- **-DIST** *dist* (default none) epicentral dist km
- **-EVDP** *evdp* (default 10.0) source depth km
- **-T** **(default true) output travel time**
- **-RAYP** **(default false) output ray param**
- **-GEOM** (default false) output geometrical spreading
- **-TS** (default true) output ray T* time
- **-P** (default true) output for P-wave
- **-sP** (default false) output for teleseismic sP
- **-SV** (default false) output for SV-wave
- **-SH** (default false) output for SH-wave
- **-pP** (default false) output for teleseismic pP
- **-M** *model_name* Model96 velocity model
- **-h** this command help
- **-?** this command help

An earth flattening approximation is used for spherical models.

Ultimately this program could be used to provide the geometrical spreading for a spherical Earth model.

8. ttimmod96

This program predicts and plots the first arrival times of the P, SV and SH arrivals for a given **model96** file[s].

The program is invoked as

ttimmod96 *[flags] model96_file[s]*
- **-XLEN** *xlen* (default 6.0) Length X-axis
- **-YLEN** *ylen* (default 6.0) Length Y-axis
- **-VRED** *vred* (default not used) reduction velocity
- **-HS** *hs* (default 0.0) Source depth km
- **-HR** *hr* (default 0.0) Receiver depth km
- **-TMIN** *tmin* (default 0.0) Minimum value of time

-TMAX *tmax* (default 100.0) Maximum value of time
-XMIN *xmin* (default 0.0) Minimum value of distance
-XMAX *xmax* (default 300.0) Maximum value of distance
-X0 *x0* (default 2.0) x-position of lower left corner
-Y0 *y0* (default 1.0) y-position of lower left corner
-K *kolor* (default 1) Profile in color. If *kolor < 0* plot models with progression from red to blue
-K kolor (default 1) Profile in color
if kolor < 0 use red->blue progression
-P (default) Plot P times
-SV (default P) Plot SV times
-SH (default P) Plot SH times
-W *width* (default 0.001") Line width in inches
-LEG (default none) Put in file legend
-?
-h Online help.

The program also creates an output file *TTIMMOD96.TXT* which has four columns of output: distance (km), P time (sec), SV time (sec) and SH time (sec). Of these only one set of travel time-distance values are plotted according to the use of the **-P**, **-SV** and **-SH** flags. The **-VRED** *vred* can change the plot to a reduced travel time plot, but this does not affect the listing of times in the file *TTIMMOD96.TXT*.

The graphics file *TTIMMOD96.PLT* is also created.

Note also that this is preliminary code and that the presentation must be cleaned up.

9. tspec96

tspec96 generates the Green's functions for TI media as a function of distance and frequency. Its output is used by **hpulse96** to yield the Green's functions as a function of distance and time. Although program will accept an isotropic or TI model specification, computation time for an isotropic model will be greater than for **hspec96** because of the need to compute eigenfunctions and eigenvalues and then the propagator matrices. **hspec96** uses known algebraic equations for the propagator matrices. The Green's functions will be the same as those from **hspec96** if the model is isotropic.

Figure 3 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

tspec96 *[flags]*, where the command flags are
-H (default false) Use Hankel function not Bessel. This will be useful at large distances and high frequencies, especially when phase velocity windowing is performed.

-A arg (default arg=3.0) value of **kr** where $\mathbf{H_n(kr)}$ replaces $\mathbf{J_n(kr)}$ in integration - only used when -H is used
-K (default Futterman) use Kjartansson Causal Q

The following govern wavefield at source. The default is the entire wavefield
-SU (default whole wavefield) Compute only upgoing wavefield from the source
-SD (default whole wavefield) Compute only downgoing wavefield from the source
-SPUP Include upward P at source
-SSUP Include upward S at source
-SPDN Include downward P at source
-SSDN Include downward S at source

The following govern the wavefield at the receiver. The default is the entire wavefield. The usefulness and effect of these for a surface receiver is not clear.

-RD Include only downgoing waves at receiver
-RU Include only upgoing waves at receiver
-RPUP Include upward P at receiver
-RSUP Include upward S at receiver
-RPDN Include downward P at receiver
-RSDN Include downward S at receiver
-?
-h Online help concerning program usage

If we run the following script,

```
hprep96 -M model.TI -d dfile -HS 10 -HR 0 -ALL
tspec96 > hspec96.out
hpulse96 -V -p -l 4 > TI.vel
fplot96 < TIvel
mv FPLOT96.PLT TIhPLOT

hprep96 -M model.IS -d dfile -HS 10 -HR 0 -ALL
tspec96 > hspec96.out
hpulse96 -V -p -l 4 > IS.vel
fplot96 < ISvel
mv FPLOT96.PLT IShPLOT
```

we will be able to compare the Green's functions for the TI and isotropic media. Figure 4 shows this comparison for the SS source Green's functions.

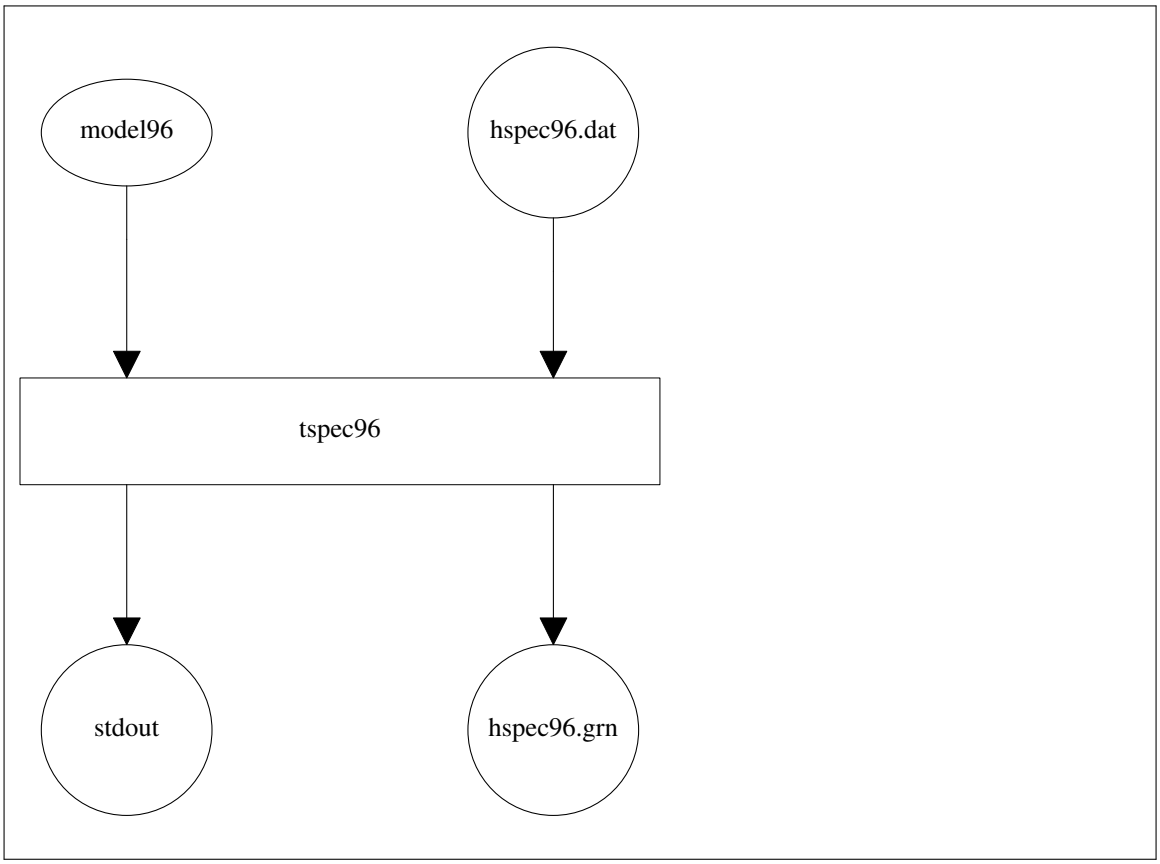


Fig. 3. Processing flow for **tspec96**

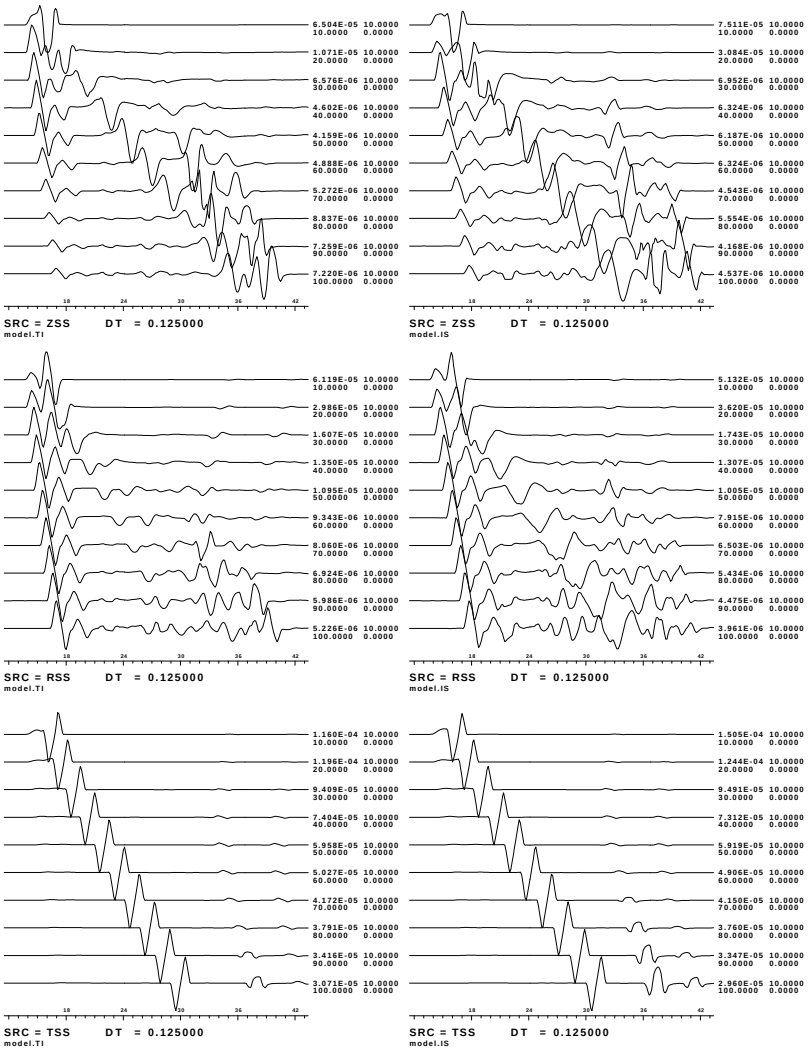


Fig. 4. Comparison of the ZSS, RSS and TSS Green's functions for the TI media (left) and equivalent isotropic media (right). Synthetics are made at distances of 10 - 100 km in increments of 10 km. Each trace consists of 256 points with **dt**=0.125 sec. A reduction velocity of 8.0 km/sec is used and each trace start at the time $-$**1.0** + **r/8.0**

10. tspec96p

tspec96p generates the Green's functions for TI media as a function of a ray parameter and frequency. Its output is used by **hpulse96** to yield the Green's functions as a function of ray parameter and time. Although program will accept an isotropic or TI model specification, computation time for an isotropic model will be greater than for **hspec96p** because of the need to compute eigenfunctions and eigenvalues and then the propagator matrices. The Green's functions will be the same as for the isotropic case.

This program computes the p-τ response for a layered media. The wavenumber integration method for computing synthetic seismograms consists of evaluating the double integral

$$g(r, t) = \int_{-\infty}^{\infty} g(r, f) e^{j2\pi ft} df$$

where

$$g(r, f) = \int_{0}^{\infty} g(k, f) J_n(kr) k dk$$

Since this equation is a Hankel transform, we also have

$$g(k, f) = \int_{0}^{\infty} g(r, f) J_n(kr) r dr$$

In exploration seismology it convenient to think in terms of ray parameter p which is related to k and $\omega = 2\pi f$ by $k = p\omega$. By substitution, we define $g(p, \tau)$ as

$$g(p, \tau) = g(p, t - pr) = \int_{-\infty}^{\infty} g(k = 2\pi fp, f) e^{j2\pi f(t-pr)} df$$

$$= \int_{-\infty}^{\infty} g(k = 2\pi fp, f) e^{j2\pi f\tau} df$$

This last expression shows that the $p - \tau$ response is actually the inverse Fourier transform of g(k, f) with the constraint $k = p2\pi f$.

The $p - \tau$ seismogram has several interesting properties. Reflection hyperbolas in the $r - t$ domain, appear as ellipses in the $p - \tau$ domain. A refracted arrival with ray parameter, p_{refr}, is mapped into a point with ray parameter p_{refr} in the $p - \tau$ domain. More importantly, the effect of the Hankel transform is that geometrical spreading is removed. The $p - \tau$ time history gives correct relative amplitudes due to plane wave reflection and refraction in the model.

Figure 5 shows the processing flow for this program. The program requires the *hspec96p.dat* file created by **hprep96p(VI)**. The program output is on *stdout* and on a binary file **hspec96.grn**.

Program control for **tspec96p** is through the command line:

`tspec96p` *[flags]*, where the command flags are

`-K` (default Futterman) use Kjartansson Causal Q

The following govern wavefield at source. The default is the entire wavefield

`-SU` (default whole wavefield) Compute only upgoing wavefield from the source

`-SD` (default whole wavefield) Compute only downgoing wavefield from the source

`-SPUP`

Include upward P at source

-SSUP
Include upward S at source
-SPDN
Include downward P at source
-SSDN
Include downward S at source
The following govern wavefield at receiver. The default is the entire wavefield
-RD Include only downgoing waves at receiver
-RU Include only upgoing waves at receiver
-RPUP Include upward P at receiver
-RSUP Include upward S at receiver
-RPDN Include downward P at receiver
-RSDN Include downward S at receiver
-?
-h
Online help concerning program usage

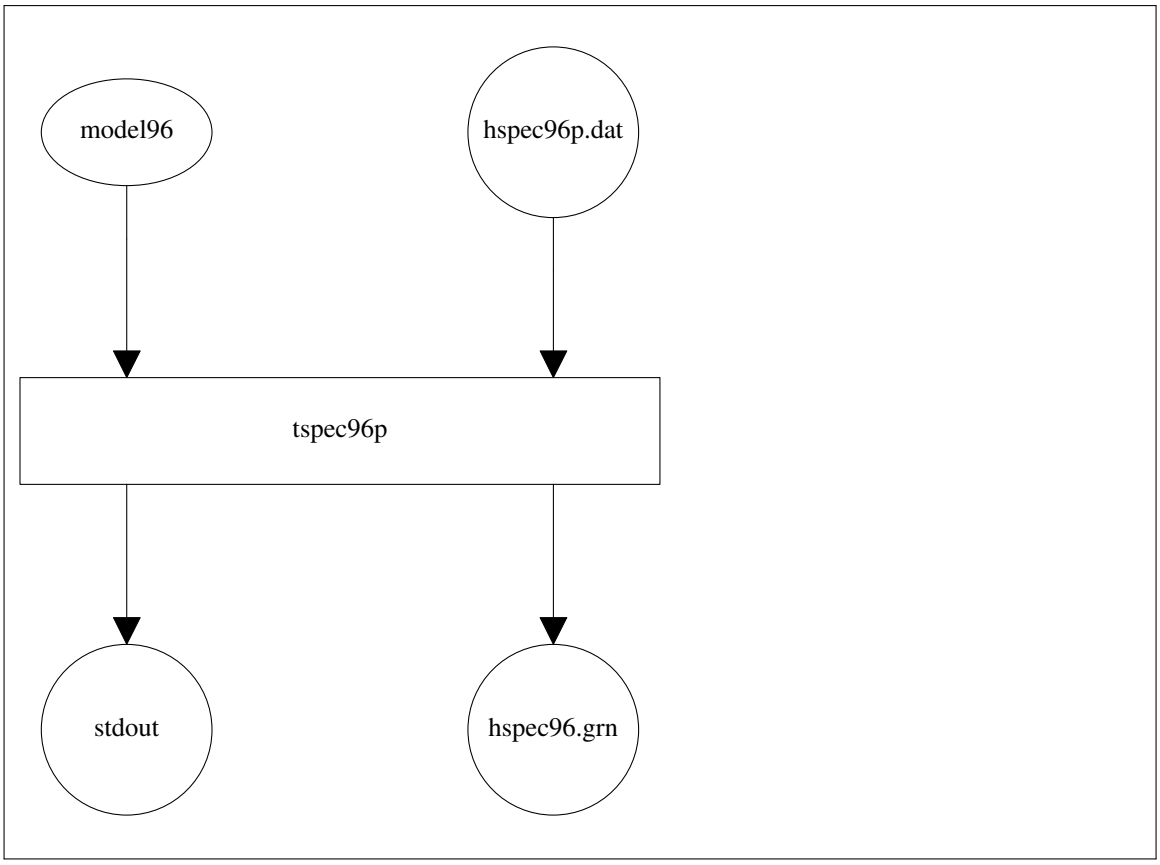


Fig. 5. Processing flow for **tspec96p**

```
rbh> hprep96p -M model.d -d dfile -HS 10 -HR 0 -ALL \
          -PMIN 0.01 -PMAX 0.20 -DP 0.01

rbh> tspec96p   > tspec96p.out

rbh> hpulse96 -V -p -l 4  | fprof96
```

11. trftn96

This program computes the vertical and radial components for incident quasi-P and quasi-SV at the top layer of a model consisting of a free surface overlying a layered half-space. The program also computes the surface receiver function. The output is a sacfile.

Program control is through the command line:

trfn96 *[flags]*, where the command flags are

```
USAGE: trftn96 [-P] [-S] [-2] [-r] [-z] -RAYP p -ALP al-
  pha -DT dt -NSAMP nsamp
-P              (default true )   Incident P wave
-S              (default false)   Incident S wave
-RAYP p         (default 0.05 )   Ray parameter in sec/km
-DT dt          (default 1.0  )   Sample interval for synthetic
-NSAMP nsamp (default 512  )   Number samples for synthetic
-M   model   (default none )   Earth model name
-ALP alp        (default 1.0  )   Number samples for synthetic
            H(f) = exp( - (pi freq/alpha)**2)
            Filter corner ~ alpha/pi
-2              (default false)   Use 2x length internally
-r              (default false)   Output radial   time series
-z              (default false)   Output vertical time series
-2              (default false) use double length FFT to
            avoid FFT wrap around in convolution
-D delay        (default 5 sec)   output delay sec before t=0
-?                      Display this usage message
-h                      Display this usage message
```

As an example of running this program consider the following which produces SAC files for the surface P-wave receiver function and surface vertical (z) and radial (r) time histories for the isotropic and TI models. The traces are plotted in Figure 6.

```
trftn96    -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.IS  -2
mv hrftn96.sac 1010.rfn
trftn96 -z   -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.IS  -2
mv hrftn96.sac z1010.rfn
trftn96 -r   -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.IS  -2
mv hrftn96.sac r1010.rfn
trftn96      -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.TI  -2
mv hrftn96.sac t1010.rfn
trftn96 -z -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.TI  -2
mv hrftn96.sac tz1010.rfn
trftn96 -r -P -ALP 1.0 -DT 0.1 -D 10. -RAYP 0.10 -M model.TI  -2
mv hrftn96.sac tr1010.rfn
```

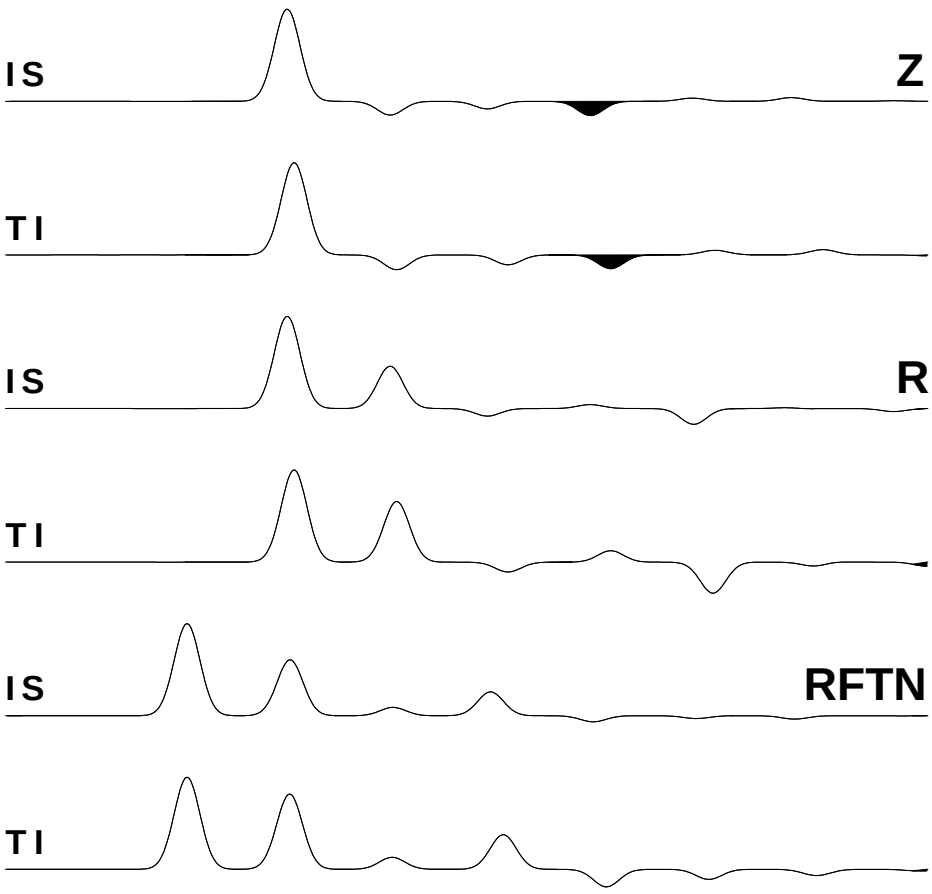


Fig. 6. Comparison of surface vertical (Z) and radial (R) synthetics and derived receiver functions (RFTN) for the isotropic and TI models. The differences become apparent at late time.

12. tfmech96

This program converts Green's functions into a three component time history for a particular source mechanism - double couple, general moment tensor or point force.

This program is in evaluation and testing.

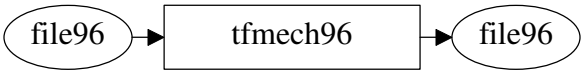
This program differs from the current **fmech96** in its support of the TI Green's functions. The determination of three-component time histories for a dislocation source for isotropic media requires only the isotropic Green's functions and the seismic moment. For TI media, the linear combination of Green's functions for a dislocation source requires terms such as **LudA** and **NudA** where **L** and **N** are two of the five TI constants, **u** is the displacement on the fault and **dA** is the fault area. For isotropic media, $\mu = L = N$ and the common term, $\mu udA \equiv M_0$ can be factored from the fault definition terms.

To preserve the useful concept of seismic moment, this program uses the value of the seismic moment, M_0 given on the command line. The program next defines

$$\mu_{eff} = \frac{1}{15}(C + A + 6L + 5N - 2F)$$

and then defines the $udA \equiv M_0 / \mu_{eff}$. Finally the exact TI relations for excitation are used since we now have a value for udA. If the source layer is isotropic then the resulting three-component time history will be the same as that from the original **fmech96**. This simplifying assumption favors the standard approach of specifying seismic moment for isotropic media while permitting the investigation of wave propagation in TI media.

An explosion in isotropic media is represented by a diagonal moment tensor with $M_{11} = M_{22} = M_{33}$. *The proper representation of an explosion in TI media is under investigation.*



Program control is through the command line:

tfmech96 *[flags]*, where the command flags are

-D *dip* dip of fault plane

-S *Strike* strike of fault plane

-R *Rake* slip angle on fault plane

-M0 *Moment (def=1.0)* Seismic moment in units of dyne-cm

-MW *mw* Moment magnitude

-E Explosion

-A *Az* Source to Station Azimuth

-B *Baz* (def=0) Station to Source azimuth

-ROT

Force the three component time histories to be vertical, radial and transverse instead of vertical, north, and east. Since the Green's functions are already vertical, radial, and transverse, the value of the back-azimuth is not used.

-fx FX -fy Fy -fZ fz

Point force amplitudes (N,E,down) in units of dynes

-XX Mxx -YY Myy -ZZ Mzz -XY Mxy -XZ Mxz -YZ Myz

Moment tensor elements in units of dyne-cm

-h

-? Online help

The program will permit a superposition of the point force with any of the moment tensor sources. however, one may wish to do this with separate operations since the source time functions of the point force and moment tensor sources may differ.

The three moment tensor specifications are mutually exclusive: e.g., strike, dip rake; explosion, moment tensor matrix. By this I mean that only one of these specified on the command line will be used to make the three component time history, in fact, the very last entry on the command line controls the result. Thus the command

```
tfmech96 -D 45 -S 120 -R 33 -E
```

will yield the three component time histories for an explosion.

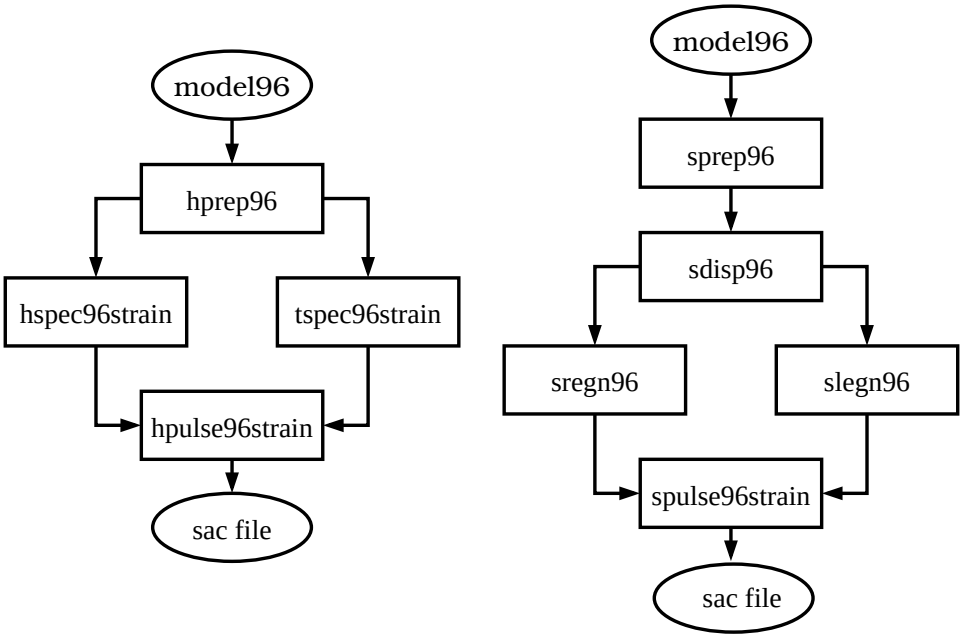
13. Limitations and Directions

The names of some of the programs may change since a TI medium includes an isotropic medium. The programs **ttimmod96**, **tshwmod96** or **tfmech96** may be renamed **timmod96**, **shwmod96** or **fmech96** respectively. The functionality of **tmkmod96** may be included in **mkmod96**. The computational programs **tspec96**, **tspec96p** and **trftn96** will not be renamed since the original programs for isotropic media are faster for computing synthetics.

CHAPTER 10
STRESS, STRAIN, DILATATION AND ROTATION
NEW

1. Introduction

This chapter describes the wavenumber integration and modal superpoistion codes to generate stress, strain, dilatation and rotation synthethics for a given specification of the force or moment of the source. The next figure shows the order that programs are run in order to go from the model to the output sac files.



The program **hprep96** was discussed in Chapter 3, while **sprep96**, **sdisp96** and **sregn96** are discussed in Chapter 4.

2. hspec96strain and tspec96strain

Figure 3 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

USAGE: **hspec96strain [-H] [-A arg] [-K] [-N][-SU] [-SD] [-SPUP] [-SSUP] [-SPDN] [-SSDN] [-RU] [-RD] [-RPUP] [-RSUP] [-RPDN] [-RSDN] [-?] [-h]**

-H (default false) Use Hankel function not Bessel
-A arg (default arg=3.0) value of kr where Hn(kr) replaces
Jn(kr) in integration - only used when -H is used
-K (default Futterman) use Kjartansson Causal Q
-N (default causal) use non-causal Q
The following govern wavefield at source. The default is the entire wavefield
-SU (default whole wavefield) Compute only upgoing wavefield from the source
-SD (default whole wavefield) Compute only downgoing wavefield from the source
-SPUP Include upward P at source
-SSUP Include upward S at source
-SPDN Include downward P at source
-SSDN Include downward S at source
The following govern wavefield at receiver. The default is the entire wavefield
-RD Only downgoing waves at receiver
-RU Only upgoing waves at receiver
-RPUP Include upward P at receiver
-RSUP Include upward S at receiver
-RPDN Include downward P at receiver
-RSDN Include downward S at receiver
-? Display this usage message
-h Display this usage message

USAGE: **tspec96strain [-H] [-A arg] [-K] [-SU] [-SD] [-SPUP] [-SSUP] [-SPDN] [-SSDN] [-RU] [-RD] [-RPUP] [-RSUP] [-RPDN] [-RSDN] [-?] [-h]**

-H (default false) Use Hankel function not Bessel
-A arg (default arg=3.0) value of kr where Hn(kr) replaces
Jn(kr) in integration - only used when -H is used
-K (default Futterman) use Kjartansson Causal Q
The following govern wavefield at source. The default is the entire wavefield
-SU (default whole wavefield) Compute only upgoing wavefield from the source
-SD (default whole wavefield) Compute only downgoing wavefield from the source
-SPUP Include upward P at source
-SSUP Include upward S at source
-SPDN Include downward P at source
-SSDN Include downward S at source
The following govern wavefield at receiver. The default is the entire wavefield
-RD Only downgoing waves at receiver
-RU Only upgoing waves at receiver
-RPUP Include upward P at receiver
-RSUP Include upward S at receiver
-RPDN Include downward P at receiver

-RSDN Include downward S at receiver
-? Display this usage message
-h Display this usage message

3. hpulse96strain

Figure 3 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

hpulse96strain:Help
USAGE:
hpulse96strain -d Distance_File [-t -o -p -i] [-a alpha]
-l L [-D|-V |A] [-F rfile] [-m mult] [-STEP|-IMP]
[-STRESS -STRAIN -ROTATE -GRN] [-FUND] [-HIGH] [-Z]
[-LAT] [-2] [-M mode] [-LOCK] -FMT ifmt
[-M0 moment] [-MW mw] [-STK stk -DIP dip -RAKE rake]
[-FX fx -FY fy -FZ fz]
[-XX Mxx ... -ZZ Mzz] [-?] [-h]
TIME FUNCTION SPECIFICATION
-t Triangular pulse of base 2 L dt
-p **Parabolic Pulse of base 4 L dt**
-p -l 1 recommended
-l L (default 1)duration control parameter
-o Ohnaka pulse with parameter alpha
-i Dirac Delta function
-a alpha Shape parameter for Ohnaka pulse
-F rfile User supplied pulse
-m mult Multiplier (default 1.0)
-STEP (default)
-IMPf11
By default the source time function is steplike. -IMP forces impulse like. **-D -IMP** is Green s function
OUTPUT FILE NAME
The format for the name of the binary output attempts to
give information on epicentral distance (km),
source depth (km), and receiver depth(km). The options are
-FMT 1 DDDDDd_HHHh_ZZZz.cmp
e.g. 005001_1234_0045.Uz
-FMT 2 DDDDDddd_HHHhhh_ZZZzzz.cmp
e.g. 00500123_123456_004578.Erf
-FMT 3 DDDDDdHHHh.grn(default)
e.g. 0050010041.ZVF
-FMT 4 DDDDdHHHh.grn

e.g. 050010045.Srz
-FMT 5 DDDdddHhhh.grn
e.g. 5001234578.Err
where D is for epicentral distance, H source depth, and
Z receiver depth. The lower case indicates the digits
to the right of the decimal place. The examples above
are for an epicentral distance is 500.123 km, source
depth 123.456 km and receiver depth 4.578 km.
OUTPUT TIMESERIES FOR SOURCE as Ur, Ut, Uz components with strain, stress optional
-D Output is ground displacement (m)
-V Output is ground velocity (default) (m/s)
-A Output is ground acceleration (m/s^2)
-STRESS (default .false.) output stress for mechanism
units are Pa, with suffix Srr, Srf, Srz, Stt, Sfz, Szz
-STRAIN (default .false.) output strain for mechanism
with suffix, Err, Erf, Erz, Eff, Efz, Ezz
NOTE the Ur, Ut, Uz components are created with **-STRESS**
or **-STRAIN** flags. The Uz is positive down.
-ROTATE (default .false.) output rotation for mechanism
with suffix, Wfz, Wrz, Wrf
-GRN (default false) Output Green;s functions
hpulse96strain -STEP -V -p -l 1 -GRN -FMT is same as
hpulse96 -V -p -l 1 | f96tosac -G . For KM,KM/S,GM/CM^3
model, output will be CM/S for moment of 1.0e+20 dyne-cm
of force of 1.0e+15 dyne
NOTE the Z component of ZDD ... ZHF is positive up

-TEST1 (default .false.) output CPS Green functions ,e.g.,
ZDS RDS ... RHF THF for use with moment tensor codes
and gsac MT command. This is equivalent to
hpulse96 -V -p -l 1 | f96tosac -G if -FMT 4 is used
with **hpulse96strain**
COMPUTATIONS
-Z (default false) zero phase
SOURCE MECHANISM SPECIFICATION
-DIP dip dip of fault plane
-STK Strike strike of fault plane
-RAKE Rake slip angle on fault plane
-M0 Moment (def=1.0) Seismic moment in units of dyne-cm
-MW mw Moment Magnitude
moment (dyne-cm) from log10 Mom = 16.10 + 1.5 Mw
For strike,dip,rake source mw or Moment must be specified
-EX Explosion
-AZ Az Source to Station Azimuth
-BAZ Baz Station to Source azimuth
-fx FX -fy Fy -fZ fz Point force amplitudes (N,E,down) in dynes
-XX Mxx -YY Myy -ZZ Mzz Moment tensor elements in units of

-XY Mxy -XZ Mxz -YZ Myz dyne-cm
The moment tensor coordinates are typically X = north Y = east and Z = down
If by accident more than one source specification is used,
the hierarchy is Mij > Strike,dip,rake > Explosion > Force

NOTE: The output units are related tot he model specification.
To have the desired units the model must be in KM, KM/S and GM/CM^3

-? Write this help message
-h Write this help message

4. spulse96strain

Figure 3 shows the processing flow for this program. The program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

spulse96strain:Help
USAGE:
spulse96strain -d Distance_File [-t -o -p -i] [-a alpha]
-l L [-D|-V |A] [-F rfile] [-m mult] [-STEP|-IMP]
[-STRESS -STRAIN -ROTATE -GRN] [-FUND] [-HIGH] [-Z]
[-LAT] [-2] [-M mode] [-LOCK] -FMT ifmt
[-M0 moment] [-MW mw] [-STK stk -DIP dip -RAKE rake]
[-FX fx -FY fy -FZ fz]
[-XX Mxx ... -ZZ Mzz] [-?] [-h]
TIME FUNCTION SPECIFICATION
-t Triangular pulse of base 2 L dt
-p Parabolic Pulse of base 4 L dt
-p -l 1 recommended
-l L (default 1)duration control parameter
-o Ohnaka pulse with parameter alpha
-i Dirac Delta function
-a alpha Shape parameter for Ohnaka pulse
-F rfile User supplied pulse
-m mult Multiplier (default 1.0)
-STEP (default)
-IMP
By default the source time function is
steplike. **-IMP** forces impulse like. **-D -IMP** is Green s function
OUTPUT FILE NAME
The format for the name of the binary output attempts to
give information on epicentral distance (km),

source depth (km), and receiver depth(km). The options are
-FMT 1 DDDDDd_HHHh_ZZZz.cmp
e.g. 005001_1234_0045.Uz
-FMT 2 DDDDDddd_HHHhhh_ZZZzzz.cmp
e.g. 00500123_123456_004578.Erf
-FMT 3 DDDDDdHHHh.grn(default)
e.g. 0050010041.ZVF
-FMT 4 DDDDdHHHh.grn
e.g. 050010045.Srz
-FMT 5 DDDdddHhhh.grn
e.g. 5001234578.Err
where D is for epicentral distance, H source depth, and
Z receiver depth. The lower case indicates the digits
to the right of the decimal place. The examples above
are for an epicentral distance is 500.123 km, source
depth 123.456 km and receiver depth 4.578 km.
OUTPUT TIMESERIES FOR SOURCE as Ur, Ut, Uz components with strain, stress optional
-D Output is ground displacement (m)
-V Output is ground velocity (default) (m/s)
-A Output is ground acceleration (m/s^2)
-STRESS (default .false.) output stress for mechanism
units are Pa, with suffix Srr, Srf, Srz, Stt, Sfz, Szz
-STRAIN (default .false.) output strain for mechanism
with suffix, Err, Erf, Erz, Eff, Efz, Ezz
NOTE the Ur, Ut, Uz components are created with **-STRESS**
or **-STRAIN** flags. The Uz is positive down.
-ROTATE (default .false.) output rotation for mechanism
with suffix, Wfz, Wrz, Wrf
-GRN (default false) Output Green;s functions
spulse96strain -STEP -V -p -l 1 -GRN -FMT 4 is same as
spulse96 -V -p -l 1 | f96tosac -G. For KM,KM/S,GM/CM^3
model, output will be CM/S for moment of 1.0e+20 dyne-cm
of force of 1.0e+15 dyne
NOTE the Z component of ZDD ... ZHF is positive up
-TEST1 (default .false.) output CPS Green functions ,e.g.,
ZDS RDS ... RHF THF for use with moment tensor codes
and gsac MT command. This is equivalent to
spulse96 -V -p -l 1 | f96tosac -G if -FMT 4 is used
with strainspulse96
COMPUTATIONS
-d Distance_File {required} Distance control file
This contains one of more lines with following entries
DIST(km) DT(sec) NPTS T0(sec) VRED(km/s)
first time point is T0 + DIST/VRED
VRED=0 means do not use reduced travel time, e.g.
500.0 0.25 512 -23.33 6.0
500.0 0.25 512 60 0.0

both have first sample at travel time of 60s
-LAT (default false) Laterally varying eigenfunctions
-2 (default false) Use double length internally
-M nmode (default all) mode to compute [9-fund,1=1st]
-Z (default false) zero phase triangular/parabolic pulse
-FUND (default all) fundamental modes only
-HIGH (default all) all higher modes only
-LOCK (default false) locked mode used
SOURCE MECHANISM SPECIFICATION
-DIP dip dip of fault plane
-STK Strike strike of fault plane
-RAKE Rake slip angle on fault plane
-M0 Moment (def=1.0) Seismic moment in units of dyne-cm
-MW mw Moment Magnitude
moment (dyne-cm) from log10 Mom = 16.10 + 1.5 Mw
For strike,dip,rake source mw or Moment must be specified
-EX Explosion
-AZ Az Source to Station Azimuth
-BAZ Baz Station to Source azimuth
-fx FX -fy Fy -fZ fz Point force amplitudes (N,E,down) in dynes
-XX Mxx -YY Myy -ZZ Mzz Moment tensor elements in units of
-XY Mxy -XZ Mxz -YZ Myz dyne-cm
The moment tensor coordinates are typically X = north Y = east and Z = down
If by accident more than one source specification is used,
the hierarchy is Mij > Strike,dip,rake > Explosion > Force

NOTE: The output units are related tot he model specification.
To have the desired units the model must be in KM, KM/S and GM/CM^3

-? Write this help message
-h Write this help message

5. srotate96

Given the stress, strain or ratation in cylindrical coordinates, **srotate96** gives the cartesnian values for a given angle between the new $\mathbf{x}_1$-axis ahd the r-axis.

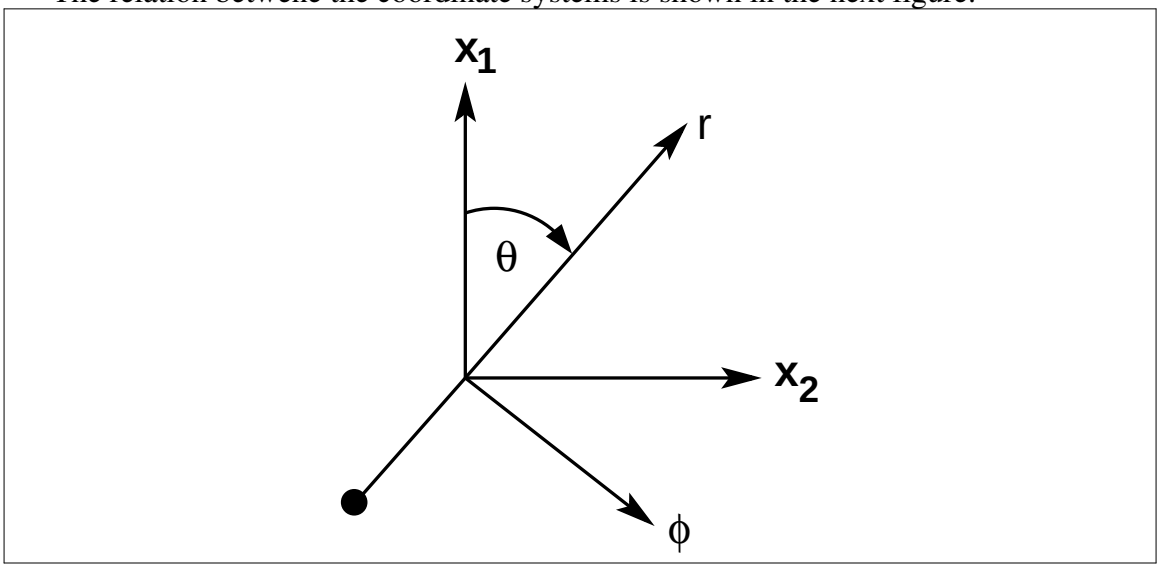
The coomand line sequence is the following:

```
srotate96 -AZ az [-U|-STRESS|-STRAIN] -FILE prototype
```

-AZ az
(required) angle between r- and x-axes
-FILE prototype (required) identifier for filename
for the example below this could be ../NEW/005000_0100_0010

-U Rotate the Ur Ut Uz from [sh]pulse96strain to Ux Uy Uz
if they exist, e.g., ../NEW/005000_0100_0010.Ur etc
to create 005000_0100_0010_Ux etc in the current directory
-STRAIN Rotate the Err Erf .. Ezz from [sh]pulse96strain to Exx Eyy ..
if they exist, e.g., ../NEW/005000_0100_0010.Err etc
to create 005000_0100_0010_Exx etc in the current directory
-STRESS Rotate the Srr Srf .. Szz from [sh]pulse96strain to Sxx Syy ..
if they exist, e.g., ../NEW/005000_0100_0010.Srr etc
to create 005000_0100_0010_Sxx etc in the current directory
-ROTATE Rotate the Wrf Wrz Wfz from [sh]pulse96strain to Wxy Wxz Wyz
if they exist, e.g., ../NEW/005000_0100_0010.Wrf etc
to create 005000_0100_0010_Wxy etc in the current directory
-h (default false) online help

The relation betwene the coordinate systems is shown in the next figure.



Transformation between the (r, ϕ, z) coordinate system to an (x_1, x_2, x_3) coordinate system. The x_3, or z coordinate is positive downward in the figure. The ϕ component of motion is the transverse component. Often the $(xsub1, x_2)$ are aligned north and east, respectively. In the case of DAS systems, one might align the x_1-axis witht he local direction of the fiber.

The transformation matrix for strain is

$$\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{12} & e_{22} & e_{23} \\ e_{13} & e_{23} & e_{33} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{rr} & e_{r\phi} & e_{rz} \\ e_{r\phi} & e_{\phi\phi} & e_{\phi z} \\ e_{rz} & e_{\phi z} & e_{zz} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This form is also used to transform the stresses.

Finally the transformation for rotations is

$$\begin{bmatrix} 0 & \omega_{12} & \omega_{13} \\ -\omega_{12} & 0 & \omega_{23} \\ -\omega_{13} & -\omega_{23} & 0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \omega_{r\phi} & \omega_{rz} \\ -\omega_{r\phi} & 0 & \omega_{\phi z} \\ -\omega_{rz} & -\omega_{\phi z} & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

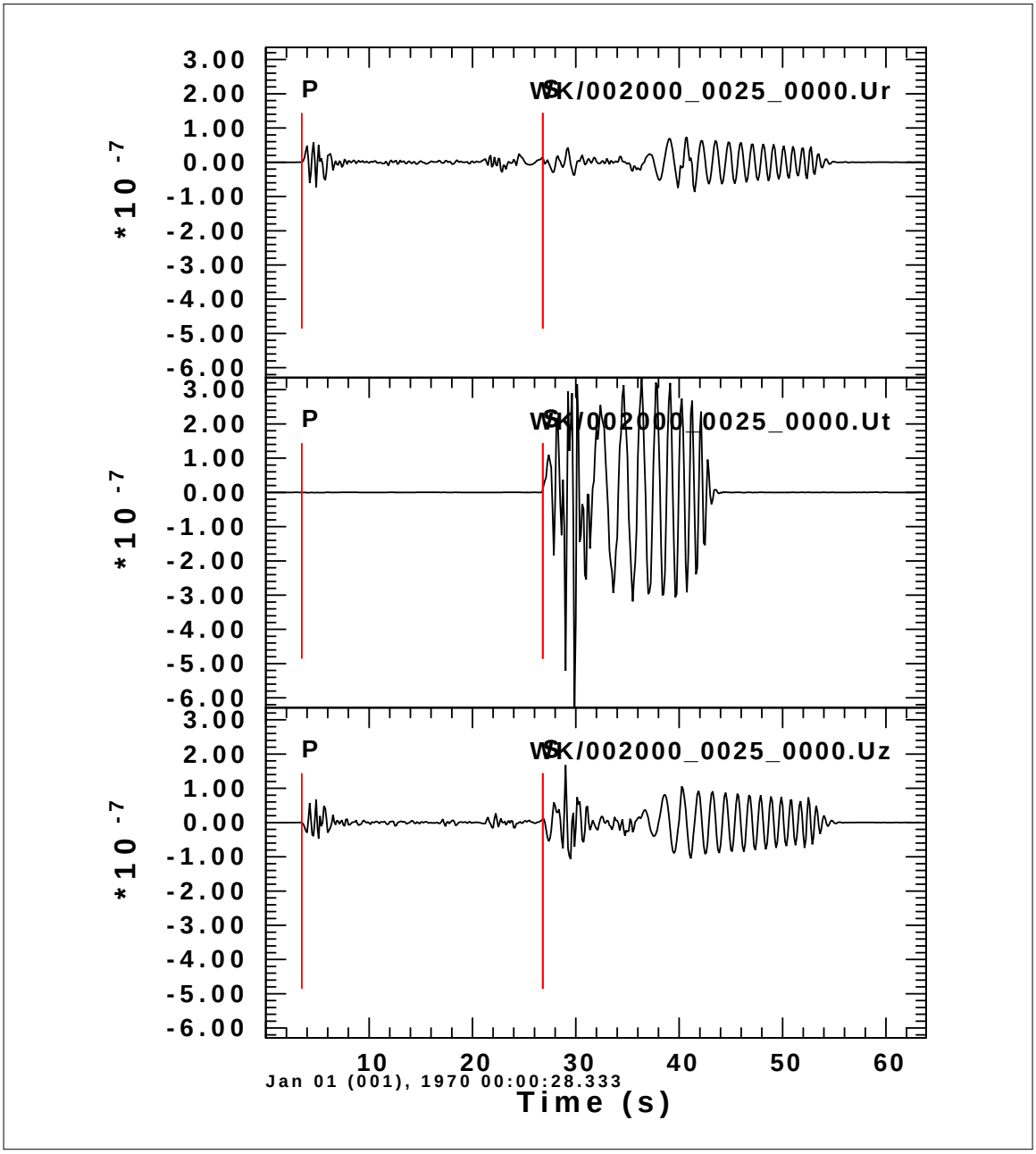
6. Sample Run

The following script is run to make the stress, strain, rotationa nd dilatation synthetics at the surface of the CUS model. The source is a strike-slip fault striking 0° at a depth of 2.5 km in the model. The observation point at 200km distance at an azimuth of 22.5°. The event has a moment magnitude of 2.6.

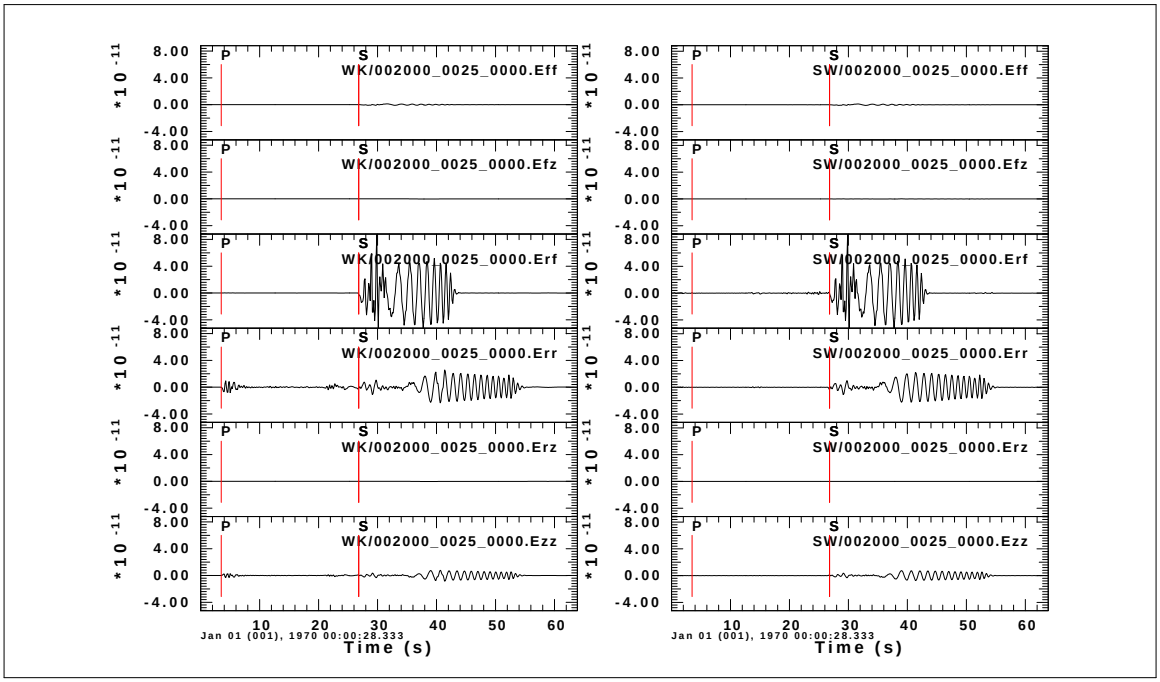
```
#!/bin/sh

cat > CUS.mod << EOF
MODEL.01
CUS Model with Q from simple gamma values
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H(KM) VP(KM/S) VS(KM/S) RHO(GM/CC)   QP   QS  ETAP  ETAS  FREFP  FREFS
  1.0000  5.0000  2.8900  2.5000 0.172E-02 0.387E-02 0.00  0.00  1.00  1.00
  9.0000  6.1000  3.5200  2.7300 0.160E-02 0.363E-02 0.00  0.00  1.00  1.00
 10.0000  6.4000  3.7000  2.8200 0.149E-02 0.336E-02 0.00  0.00  1.00  1.00
 20.0000  6.7000  3.8700  2.9020 0.000E-04 0.000E-04 0.00  0.00  1.00  1.00
  0.0000  8.1500  4.7000  3.3640 0.194E-02 0.431E-02 0.00  0.00  1.00  1.00
EOF

DIST=200
T9=0.0
HS=2.5
HR=0.0

cat > dfile << EOF
${DIST} 0.125 512 -5 6
EOF

hprep96 -d dfile -M  CUS.mod -HS ${HS} -HR ${HR}  -TF -BH  -ALL
hspec96strain

STK=0
DIP=90
RAKE=0
AZ=22.5
MW=2.6

hpulse96strain -V -p -l 1 -STRAIN -STRESS -ROTATE -MW ${MW} \
      -STK ${STK} -DIP ${DIP} -RAKE ${RAKE}  -AZ ${AZ}
mkdir WK
mv *.Del  *.U? *.[EWS]??  WK

sprep96 -d dfile -L -R -M CUS.mod -NMOD  100 -HS ${HS} -HR ${HR}
sdisp96
sregn96
slegn96
spulse96strain -d dfile -V -p -l 1 -STRAIN -STRESS -ROTATE \
   -MW ${MW} -STK ${STK} -DIP ${DIP} -RAKE ${RAKE}  -AZ ${AZ}
mkdir SW
mv *.Del *.U? *.[EWS]?? SW

DOPLT

rm -f hspec96.??? lunk sdisp96.??? *.egn
```

The **DOPLT** script uses **gsac** to make the plts shown below.

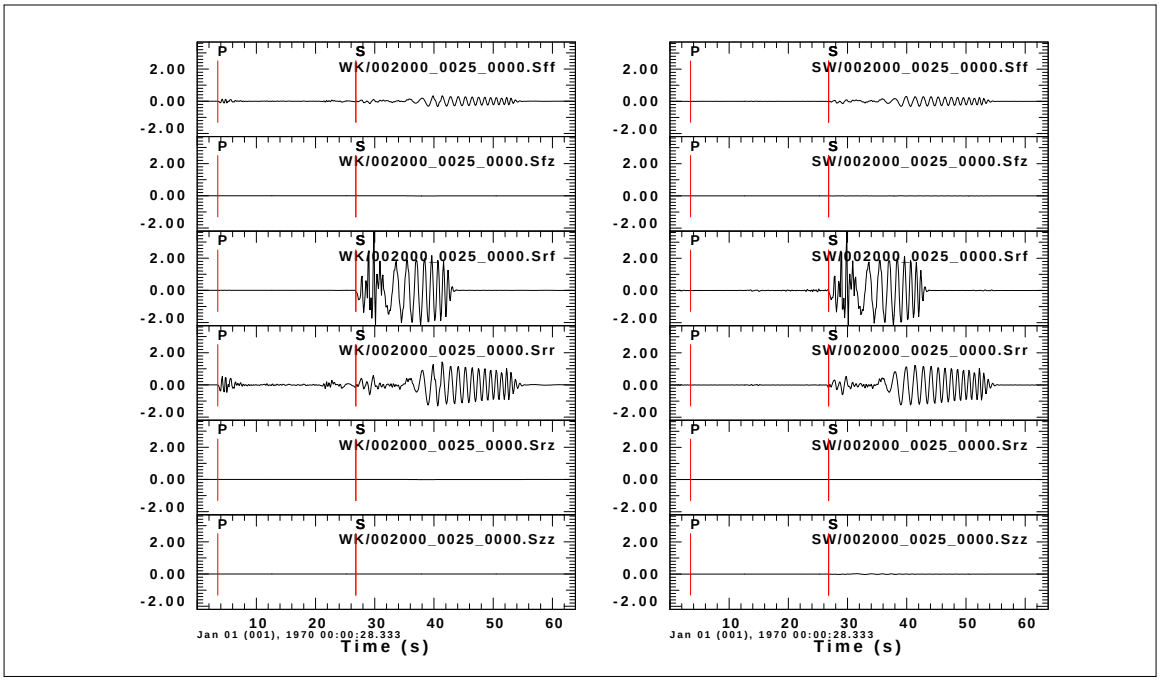
```
#!/bin/sh
gsac << EOF
ylim all
pctl xlen 4.0 x0 1.0
r WK/*.E??
fileid name
bg plt
hold on
p
pctl x0 6.0
r SW/*.E??
p
hold off
q
EOF
plotnps -F7 -W10 -EPS -K < P001.PLT > E.eps
gsac << EOF
ylim all
pctl xlen 4.0 x0 1.0
r WK/*.U?
fileid name
bg plt
hold on
p
pctl x0 6.0
r SW/*.U??
p
hold off
q
EOF
plotnps -F7 -W10 -EPS -K < P001.PLT > U.eps
gsac << EOF
ylim all
pctl xlen 4.0 x0 1.0
r WK/*.S??
fileid name
bg plt
hold on
p
pctl x0 6.0
r SW/*.S??
p
hold off
q
EOF
plotnps -F7 -W10 -EPS -K < P001.PLT > S.eps
gsac << EOF
ylim all
pctl xlen 4.0 x0 1.0
r WK/*.[WD]??
fileid name
bg plt
hold on
p
pctl x0 6.0
r SW/*.[WD]??
p
hold off
q
EOF
plotnps -F7 -W10 -EPS -K < P001.PLT > WD.eps
mv [ESU].eps WD.eps ..
```



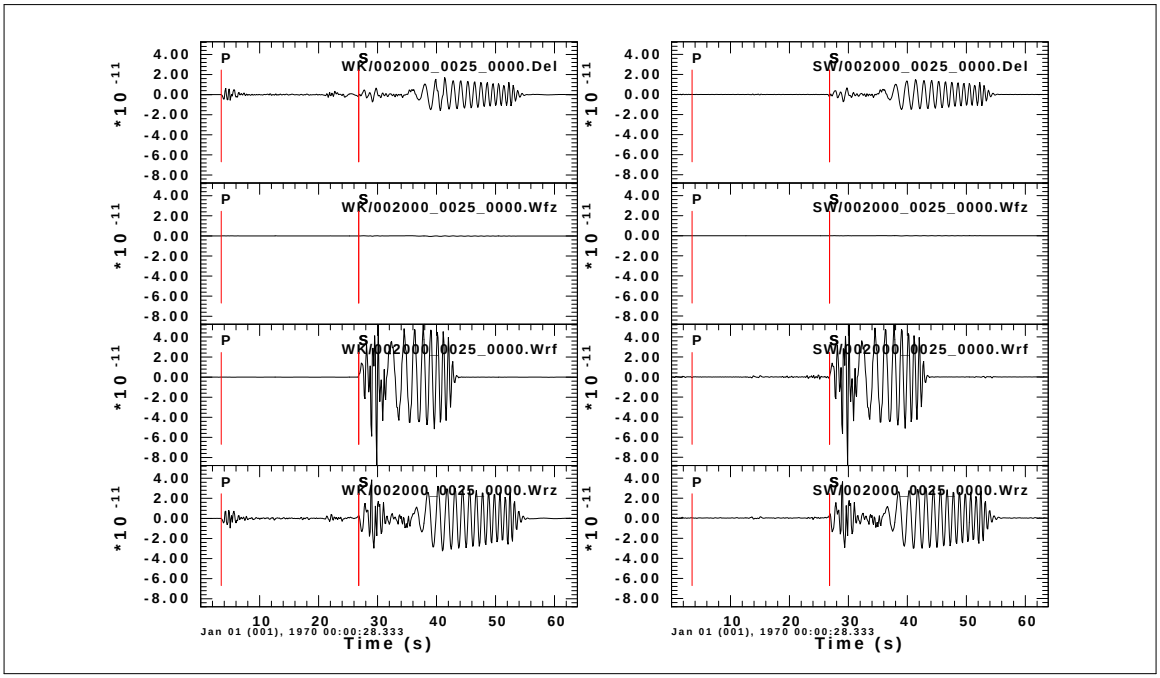
Three component velocity in m/s in the R, T and Z directions. For this program, Uz is positive down.



Three component strain in cylindrical coordinates. The 'f' indicates the ϕ component. the Efz and Erz must be zero because of the free surface stress-free condition.



Three component stress in Pa. Since the receiver is at the surface, Srz and Srz are zero.



Three component dilatation (Del) and rotations.

7. hwhole96strain NEW

This program computes the Green's functions and their vertical and radial derivatives in an isotropic wholespace by using analytic expression for the frequency response. Its purpose is to provide an independent solution to which **hspec96strain** can be compared. It serves the same function as **hwhole96** does to **hspec96**, e.g., to check the wavenumber integration at low frequencies when the epicentral distance is large compared to the vertical separation between the source and receiver.

The processing flow for this program is the same as that of Figure 3, except that the **hwhole96strain** is run instead of **hspec96strain**.

This program requires the *hspec96.dat* file created by **hprep96(VI)**. The program output is on *stdout* and on a binary file *hspec96.grn*.

Program control is through the command line:

USAGE: **`hwhole96strain [-K] [-?] [-h]`**

`-K` (default Futterman) use Kjartansson Causal Q

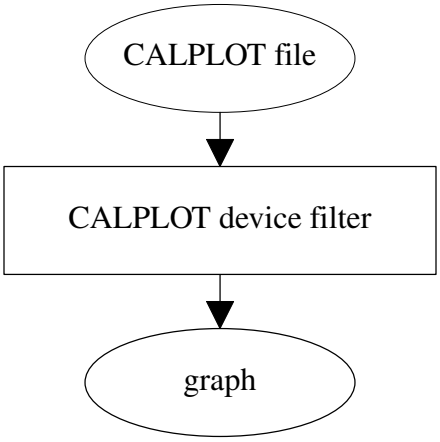
`-?` Display this usage message

`-h` Display this usage message

APPENDIX A
CALPLOT GRAPHICS (REVISED)

1. Introduction

Computer Programs in Seismology is distributed with its own graphics package to make the installation of distributed software easier. Each non-interactive graphics program will create a binary, device-independent metafile of plotting commands, which must be converted for use by a specific hardware device. At the simplest level, the low level plotting commands are a sequence of pen up, pen move and pen down commands. Some of the early plotting devices supported were Calcomp mechanical plotters, Versatec electrostatic printers and Tektronix graphics terminals. Today graphic output is supported for X-Windows, PC Windows displays, and PostScript printers. Only a small subset of output devices are currently supported, primarily because of the existence of excellent conversion software; one example of which is *ghostscript* which converts PostScript to many devices.



If a CALPLOT device filter is named **plotdev**, one uses the program as follows:

`plotdev [options] < PLOTFILE` *for a screen device*

`plotdev [options] < PLOTFILE > temp_file` *(create temp file)*
`print temp_file` *(output to the actual printer)*

Some common options are

`-S`*scaling_factor*
Multiply all plot moves by the *scaling_factor* (default 1.0)
`-R`
Rotate the plot by 90°
`-F`*font*

Change the default fault to number *font*. The default is Times Roman. A **`-F7`** will invoke bold Helvetica in PostScript.

Other commands are specific to the hardware device. A complete description of all supported devices is given in **CALPLOT(I)** of *Computer Programs in Seismology*.

2. PostScript Output

The program **plotnps** converts the binary CALPLOT file to PostScript. This program supports 128 unique colors in its palette. The output can also be in the form of Encapsulated PostScript, which is used to provide all graphics in this document.

Program control is through the command line:

`plotnps` *[flags]*, where the command flags are

`-S`*scalefac*
Scale all plot motions by this factor.

`-P`*pipe_process*
On UNIX/LINUX pipe the PostScript output through this process instead of sending through *stdout*

`-R`
Rotate the plot on the printed page. In effect the plot region is 8.5" wide and 11.0" high instead of 11.0" wide and 8.5" high.

`-N`
Turn off shading options for smaller PostScript file

`-F`*font*
Make the default font equal *font*

font	Font Used
0	Times-Roman
1	Times-Roman
2	Times-Italic
3	Times-Bold
4	Symbol (Greek)
5	Helvetica
6	Helvetica-Oblique
7	Helvetica-Bold
8	Symbol (Greek)
9	Courier
10	Courier-Oblique
11	Courier-Bold
12	Symbol (Greek)

Shading commands:

-H30
-H60
Use a halftone density of 30 or 60 (default) dots per inch. The density of 30 produces larger dots, and may be of use when a figure must be reduced for publication. This is old. Use -G fro grayscale

-K
Show colors with a red -> green -> blue palette

-KR
Show colors with a red -> white -> blue pallette

-KB
Show colors with a blue -> white -> red pallette

-KW
Show colors, but whiten the spectrum.

-G
Show colors in grayscale.

The default action when neither **-G**, **-K** nor **-KW** are used is that shading is in gray, but all colored lines and text are black.

-I (default off) Invert background (e.g make black) for EPS

-BGFILL (default off) Force a background fill for use with ImageMagick or GraphicsMagick

Media commands:

-B
assume the paper is 11 x 14 instead of 8.5 x 11

-L
assume the paper is 8.5 x 14 instead of 8.5 x 11

-A3
assume the paper is A3 instead of 8.5 x 11

-A4
assume the paper is A4 instead of 8.5 x 11

-W*min_linewidth*
Reset the minimum line width.

-EPS
Make the output an Encapsulated PostScript file.

-T*title*
Place the title string *title* in the lower left corner. Do not use spaces, or under UNIX/LINUX place string between quotes, e.g., **-T***"a test case"*

-X0xoff (default 0) x-offset in CALPLOT units

-X0xoff (default 0) y-offset in CALPLOT units

-h

-? Online help

Standards

To be compatible with PostScript display software and with word processing software that permits inclusion of PostScript files, PostScript Document Structure Convention 3.0 (DSC 3.0) is followed.

Plotspace Mapping

The CALPLOT definition of axes is such that the X-axis is horizontal and the Y-axis is vertical. This is then mapped onto a printed page of dimension 8.5" x 11". In the default case the X-axis is mapped onto the long dimension of the paper. The plot space on the paper is demonstrated in Figure 1.

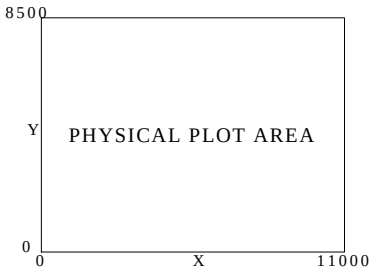


Fig. 1. Default mapping of CALPLOT plot space onto physical page.

The **-R** option rotates the mapping, such that the Y-axis is mapped onto the long dimension of the paper. The plot space on the paper is demonstrated in Figure 2.

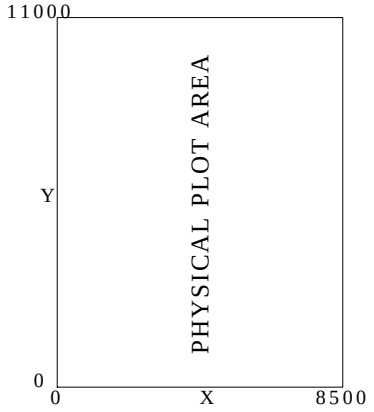


Fig. 2. Mapping of CALPLOT plot space onto physical page using *-R*.

Note that the CALPLOT plot space is mapped onto a rectangular page with no distortion of the unit lengths of the X- or Y-axes.

The PostScript plot space is assumed to be that the X-axis is horizontal with a length of 8.5" and the Y-axis is vertical with a length of 11.0". The use of the **-EPS** or **-LEPS** options permits a plot to be able t0 fit within these limits. For the **-EPS** option, the CALPLOT X-axis will still be horizontal. The default and **-R** option changes the lengths of the plotted axes in the manner consistent with Figures 1 and 2. The **-LEPS** option make the X-axis parallel to the long direction of the page.

The following examples use the second page of the example file *PLTTST* to illustrate the result of using this program with different options.

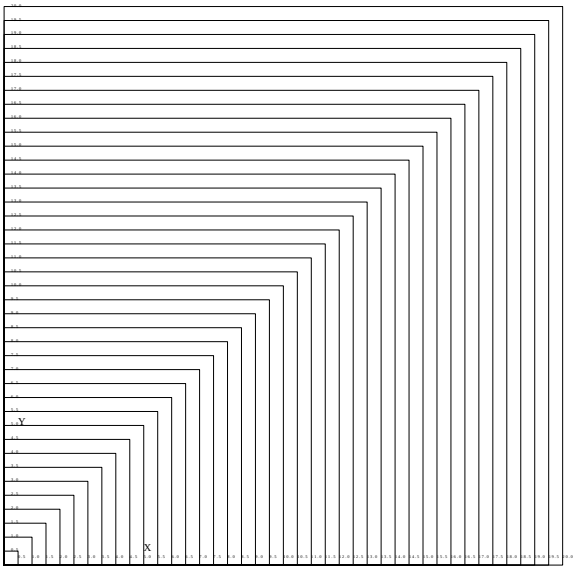


Fig. 3. This is the result of using *plotnps -EPS < plt > plttst.eps*

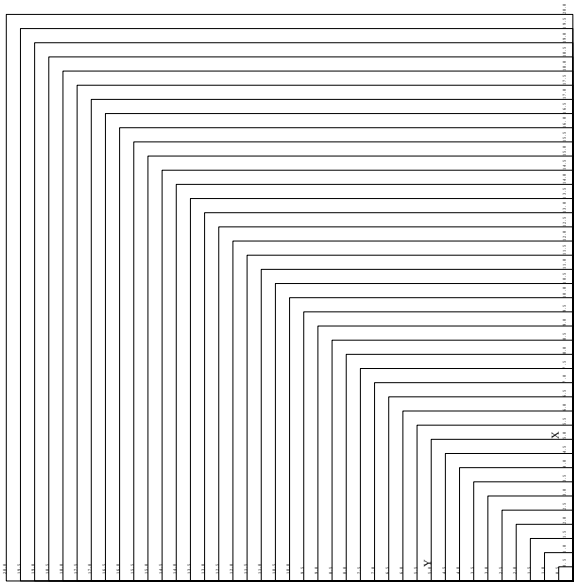
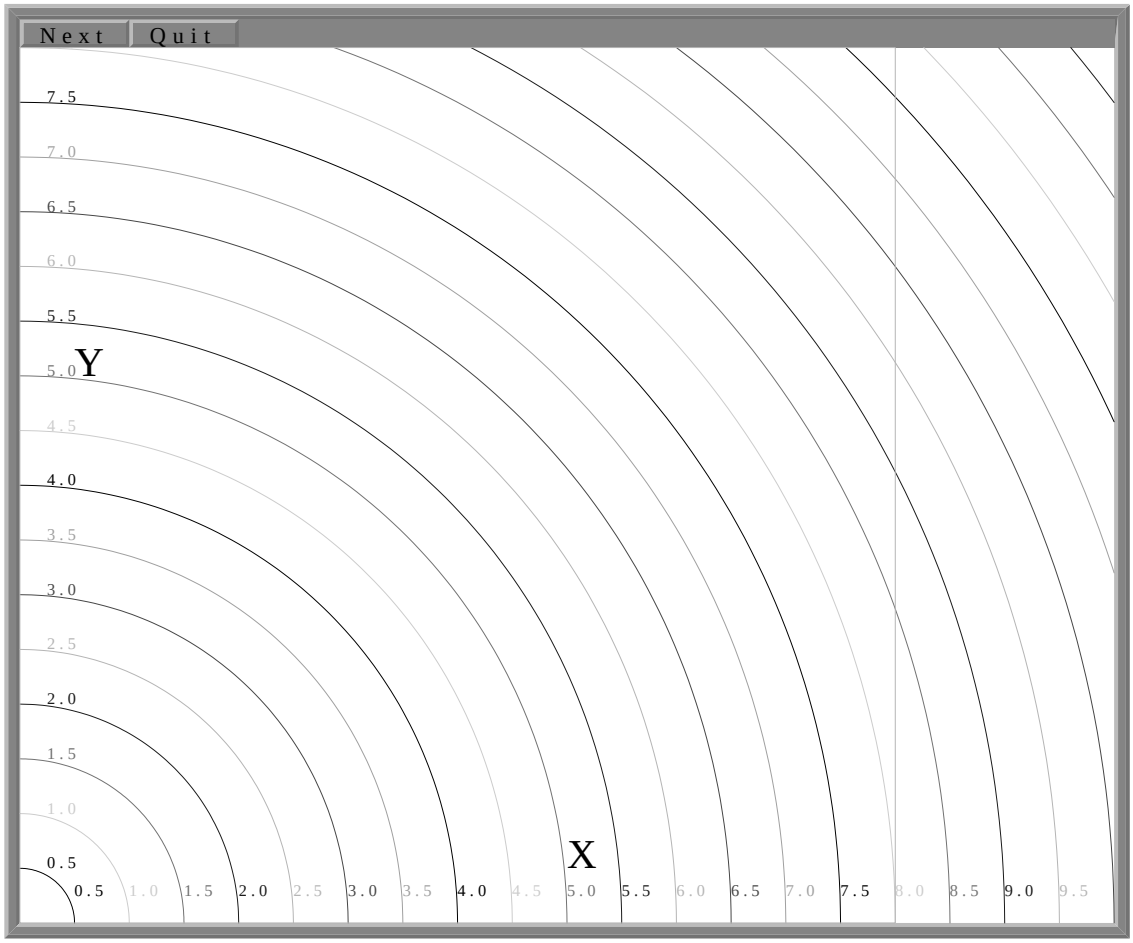


Fig. 4. This is the result of using *plotnps -EPS -R < plt > plttst.rps*

The Windows screen is viewed as a piece of paper exactly 10.0" wide and 8.0" high (approximately 25.4 cm wide by 20.32 cm high). The default screen has dimensions of 800 x 640 pixels.



When the page is completely drawn, a cursor will appear. One can use this to point to a feature of interest. The following actions can be performed:

- Pressing and releasing the *Left Mouse Button* will advance the page.
- Pressing the menu button *Next* will advance the page.
- Pressing the menu button *Quit* terminates the plot.

In order of importance, an entry such as this overrides a command line or environment option. For the other options, the command line overrides the PLOTMSW environment control.

Last Modified 05 NOV 2001

3. X11 Output

The program **plotxvig** is a native X11 program for use under the X11 windowing system. It is based on the XviG Version 1.1 package (Antoon Demarrée, IMEC, © 1993). This program supports 35 unique colors in its palette. If these colors are not available, dithering is used to create the apparent set. This package is also the basis of interactive

X11 software.

Program control is through the command line:

plotxvig *[flags]*

-S*scalefac*
Scale all motions by this factor.

-F*font* Change the default font.

-R Rotate the plot by 90°

-N No shading

-I Invert the background. The background will be black instead of white. This is done by interchanging the black and white color map entries

-K (default) Show colors with a red -> green -> blue pallette

-KR Show colors with a red -> white -> blue pallette

-KB Show colors with a blue -> white -> red pallette

-G Grayscale *(color is default)*

-W*width* Minimum linewidth in units of 0.001" or 0.0025cm

-geometry width*x*height+-xoff+-yoff
set geometry in manner of X11. The *xoff* and *yoff* are optional. *(Default width=800, height=640).*

-p

-p2

-p4

-p10

-C Turn on cross-hairs. This is useful with the *-p* options to define crop limits when using **reframe**.

-h Online help
Put up background positioning grid every 1000 CALPLOT units. every 500, every 250 or every 100, respectively

To provide additional user control, the command line arguments can be placed in the environment by separating them by colons (:) with no intervening spaces. **This is the only way to change display options when using the graphics libraries are used for interactive plots**. To force a scale factor of 0.5, and the images size of 800x600 one would set the environment parameter **PLOTXVIG**

```
set PLOTXVIG=:-S0.5:-g:800x600:
export PLOTXVIG        (under sh or ksh)
```

or

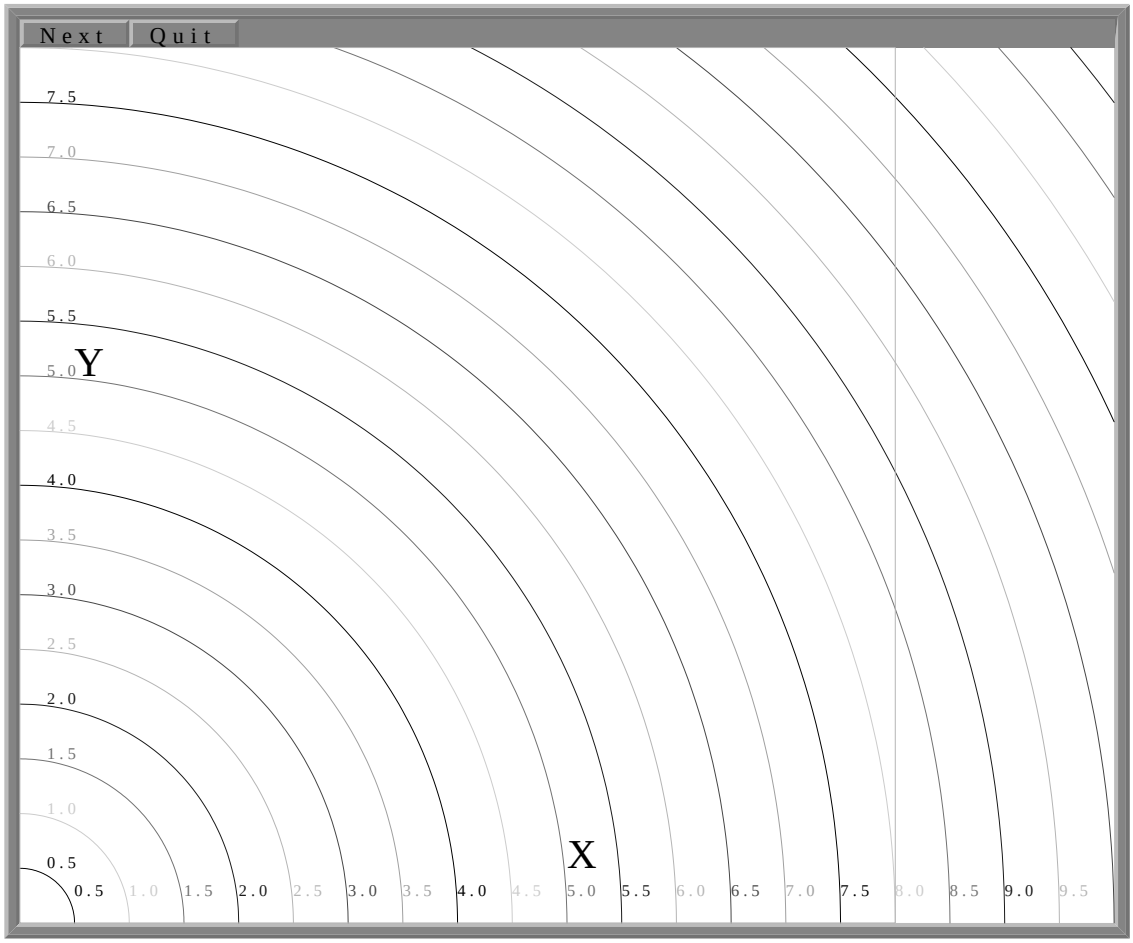
```
setenv PLOTXVIG=:-S0.5:-g:800x600:    (under csh)
```

-h

-?
Online help

The X11 screen is viewed as a piece of paper exactly 10.0" wide and 8.0" high (approximately 25.4 cm wide by 20.32 cm high). The default screen has dimensions of 800 x 640

pixels, which can be changed through the window manager when the program begins. The following screen would appear:



When the page is completely drawn, a cursor will appear. One can use this to point to a feature of interest. The following actions can be performed:

- Pressing and releasing the *Left Mouse Button* will advance the page.
- Pressing the menu button *Next* will advance the page.
- Pressing the menu button *Quit* terminates the plot.

To be consistent with X11, the geometry of the plot window can be specified by an entry in the .Xdefaults file:

```
plotxvig.calxvig.plotxvig.geometry: 1000x800+100+50
```

In order of importance, an entry such as this overrides a command line or environment option. For the other options, the command line overrides the PLOTXVIG environment control.

Problems:

Since X11 programming is a new experience, here are some annoyances.

Resizing a window after plotting begins will truncate the plot, if the window is smaller, or will have unused areas. Because of the size of the binary plotfiles, there is no way to rewind and redraw a plot. Instead a backup image is used.

Finally, **`plotxvig`** works by setting up two UNIX processes: one to do do the drawing, the other to handle events and to place the drawing on the screen - an interesting use of interprocess communication. You may find yourself with a display that is not responsive - this usually happens because one, but not both processes have terminated. Use **ps** to list the processes, and then **kill `PID`** to get rid of **calxvig** and **plotxvig**.

4. Figure Manipulation

The program **reframe** permits manipulation of a CALPLOT figure, either by changing the position on the page, by imposing a primitive clipping. Options exist to select one figure of a multipage plot file, and to merge plot files. The output of this program is another plot file. The program imput is from the last argument on the command line, if that argument is not a command flag, or the standard input

Program control is through the command line:

`reframe` *[flags]*, where the command flags are

`-O`
: Redirect the output to the standard output. Otherwise a **`plotXXXXX`** file will be created, where **XXXXX** is a unique identification number.

`-P`
: Force the output to be a plot file. This the default.

`-M`*mergefile*
: This is the file that will be superimposed onto the original file.

`-XL`*x_low_clip* (default = -100000000)
`-XH`*x_high_clip* (default = 100000000)
`-YL`*y_low_clip* (default = -100000000)
`-YH`*y_high_clip* (default = 100000000)
: A selected position of the input figure can be passed through to the output. The selected region is bounded by these coordinates.

`-X0`*x_origin*
`-Y0`*y_origin*
: These values are added to the (x,y) coordinates of all input values within the clipping window to shift the resulting figure on the page.

  The sequence of operations is that first the image is clipped, and then the origin is shifted.

To illustrate the usage of the program, consider the following two examples:

To merge the second frames of two plot files, one need only do

```
reframe -N2 -O -MPLOTrhwvint < PLOTrefplt > PLOTrefplt2
```

Note that the page number flag applies to both input files. It may be necessary to run the program three times to select the desired pages, first two runs, and then to merge the output using the temporary files.

To select the first three pages of a multipage plotfile, and then to combine them on to a single page,

```
reframe -V  -XH8500 -YH11000  -N1 -O < TABL > hunk1
  ( retrieve page 1 and save in the file hunk1)
reframe -V  -XH8500 -YH11000  -X0+8750  -N2 -O < TABL > hunk2
  ( retrieve page 2, move plot 8750 units to right and
     save in the file hunk2)
reframe -V  -XH8500 -YH11000  -X0+17500 -N3 -O < TABL > hunk3
  ( retrieve page 3, move plot 11000 units to right and
     save in the file hunk3)
reframe -V -N1 -O -Mhunk2 hunk1 > munk1
  ( merge the files hunk2 and hunk1 into the file munk1)
reframe -V -X0+1000 -Y0+1000 -N1 -O -Mhunk3 munk1 > PLOTreframe2
  ( merge files hunk3 and munk1, and shift the origin 1000 units to
     the right and upward)
```

All units are in the device independent plot units. When the CALPLOT programs are used, 1000 units correspond to 1.000 inches on the hardcopy plot.

The results of another example are shown in Figures 3 and 4. The object is to cut Figure 3 into four quadrants centered at (4.0,4.0) and to exchange the upper right with the lower left quadrant and the upper left with the lower right quadrant. The commands used are as follow:

```
reframe -N1 -O < PLTTST > p
reframe -N1 -O -X0+4000 -Y0+4000 -XL0000 -XH4000 -YL0000 -YH4000 < p1 > g1
reframe -N1 -O -X0-4000 -Y0+4000 -XL4000 -XH8000 -YL0000 -YH4000 < p1 > g2
reframe -N1 -O -X0+4000 -Y0-4000 -XL0000 -XH4000 -YL4000 -YH8000 < p1 > g3
reframe -N1 -O -X0-4000 -Y0-4000 -XL4000 -XH8000 -YL4000 -YH8000 < p1 > g4
reframe -N1 -O -Mg1 < g2 > g5
reframe -N1 -O -Mg3 < g4 > g6
reframe -N1 -O -Mg5 -X0+1000 -Y0+1000 < g6 > g7
plotnps -F7 -W10 -G -EPS < g7 > g7.eps
rm p1 g1 g2 g3 g4 g5 g6 g7
```

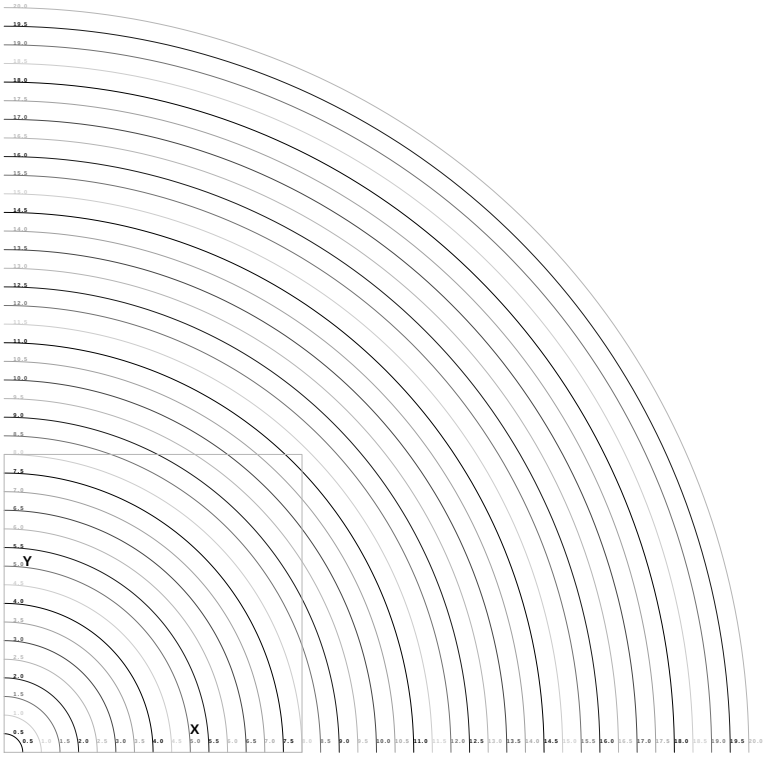


Fig. 3. Initial plot to be sectioned

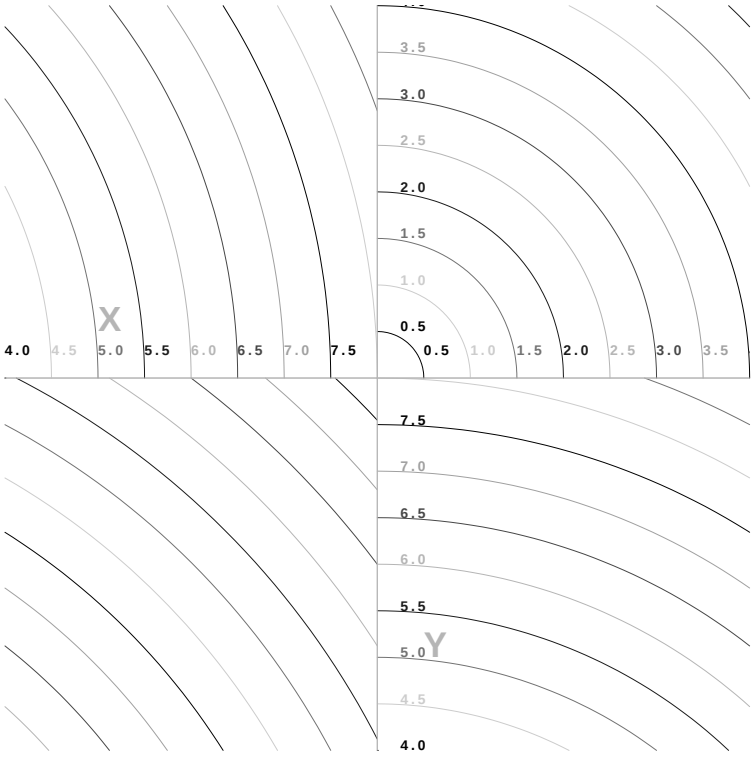


Fig. 4. Result of clipping and shifting

5. CALPLOT Colors

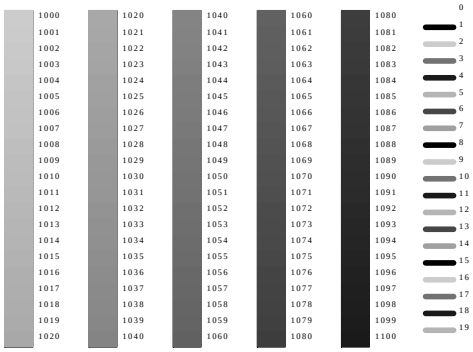
Many programs permit the user definition of colors for curves. These are invoked using

-K kolor

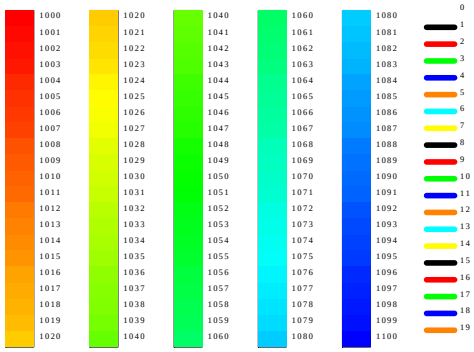
The CALPLOT graphics uses a set of predefined colors that take on slightly different meanings depending upon whether the plot program (**plotxvig, plotmsw, plotnps, plotgif**), is invoked with the **-G**, **-K**, **-KR** or **-KB** flags. Values of kolor between **0** and **999** are mapped into a specified sequence of 7 colors. Values in the range **1000** - **1100** select a palette of continuous color tones selected by the use of these flags. The table below defines some of these values as do the figures, which are best viewed on a color terminal screen using GhostView or Acroread.

Kolor	-G	-K	-KR	-KB
0	Background	Background	Background	Background
1	Foreground	Foreground	Foreground	Foreground
2	(see below)	Red	(see below)	(see below)
3		Green		
4		Blue		
5		Orange		
6		Blue-Green		
7		Yellow		
8	Foreground	Foreground	Foreground	Foreground
9		Red		
999				
1000	Lt. Gray	Red	Red	Blue
1025		Orange	Light Red	Light Blue
1050	Med.Gray	Green	White	White
1075		BlueGreen	Light Blue	Light Red
1100	Dk. Gray	Blue	Blue	Red

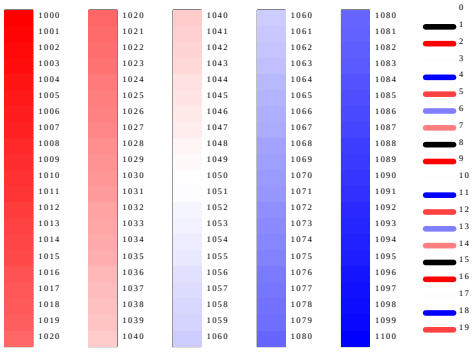
Normal plotting uses the **`-G`** and **`-K`** flags. Displays of continuous color maps can use the **`-KR`** and **`-KB`** modes if the color indices are programed to represent a range of negative - positive values with white representing a median value. The following figures show the resulting colors for a given choice of the `kolor` index.



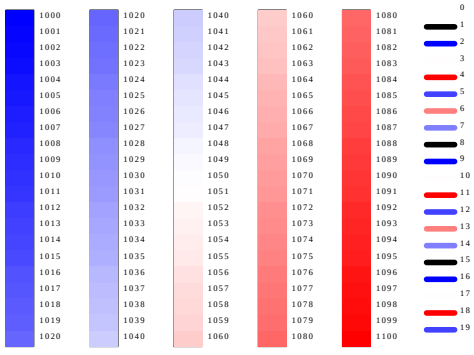
plotnps -G < GRAYSC



plotnps -K < GRAYSC



plotnps -KR < GRAYSC



plotnps -KB < GRAYSC

APPENDIX B
GREEN'S FUNCTIONS

1. Introduction

The Green's functions generated by synthetic seismogram codes can be combined to create three component time histories for arbitrarily oriented point forces, for double couple sources and for general moment tensor sources. The correspondence between the internal representation and what the represent is indicated by the following table:

Position	Green's Function	Description
1	ZDD	Vertical Component 45° dip slip
2	RDD	Radial Component 45° dip slip
3	ZDS	Vertical Component 90° dip slip
4	RDS	Radial Component 90° dip slip
5	TDS	Tangential Component 90° dip slip
6	ZSS	Vertical Component vertical strike slip
7	RSS	Radial Component vertical strike slip
8	TSS	Tangential Component vertical strike slip
9	ZEX	Vertical Component Explosion
10	REX	Radial Component Explosion
11	ZVF	Vertical Component Downward Vertical Force
12	RVF	Radial Component Downward Vertical Force
13	ZHF	Vertical Component Horizontal Force
14	RHF	Radial Component Horizontal Force
15	THF	Tangential Component Horizontal Force
16	PEX	Pressure in Fluid for Explosion
17	PDD	Pressure in Fluid for DD source in solid
18	PDS	Pressure in Fluid for DS source in solid
19	PSS	Pressure in Fluid for SS source in solid
20	PVF	Pressure in Fluid for VF source in solid
21	PHF	Pressure in Fluid for HF source in solid

2. Combinations of Green's Functions

Double-Couple Source

For an arbitrarily oriented double couple without moment source model with vector n $= (n_1, n_2, n_3)$ normal to the fault and f $= (f_1, f_2, f_3)$ in the direction of the dislocation (Haskell, 1963; Haskell, 1964), equation 11 of Wang and Herrmann (1980) for the Fourier transformed displacements at the free surface at a distance r from the origin becomes

$$\begin{aligned} u_z(r,0,\omega) &= ZSS[(f_1n_1 - f_2n_2)\cos 2\phi + (f_1n_2 + f_2n_1)\sin 2\phi] \\ &+ ZDS[(f_1n_3 + f_3n_1)\cos\phi + (f_2n_3 + f_3n_2)\sin\phi] \\ &+ ZDD[f_3n_3] \end{aligned} \tag{1a}$$

$$\begin{aligned} u_r(r,0,\omega) &= RSS[(f_1n_1 - f_2n_2)\cos 2\phi + (f_1n_2 + f_2n_1)\sin 2\phi] \\ &+ RDS[(f_1n_3 + f_3n_1)\cos\phi + (f_2n_3 + f_3n_2)\sin\phi] \\ &+ RDD[f_3n_3] \end{aligned} \tag{1b}$$

$$\begin{aligned} u_\phi(r,0,\omega) &= TSS[(f_1n_1 - f_2n_2)\sin 2\phi - (f_1n_2 + f_2n_1)\cos 2\phi] \\ &+ TDS[(f_1n_3 + f_3n_1)\sin\phi - (f_2n_3 + f_3n_2)\cos\phi] \end{aligned} \tag{1c}$$

The vertical displacement u_z is positive upward, the radial displacement is positive away from the source, and the tangential displacement u_ϕ is positive in a direction clockwise from north. The vectors n and f are still defined in a local coordinate system at the source in which the 1, 2, and 3 cartesian axes are in the north, east and downward directions, respectively. Following Herrmann (1975) the components of these vectors can be expressed in terms of the fault plane parameters of strike, dip and slip. The strike, ϕ_f, is measured clockwise from north, the dip, d_f, is measured in a positive sense from the horizontal direction perpendicular to strike, and the slip, λ_f, is measured on the fault plane in a counterclockwise sense from the horizontal direction of strike. With these conventions, all possible fault planes are encompassed by the ranges in the angles of $0^o \le \phi_f < 360^o$, $0^o \le d_f \le 90^o$, and $-180^0 \le \lambda_f < 180^o$. With this notation, the sense of P-wave first motion at the center of the focal sphere is positive for positive values of λ_f and negative for negative values. The components of the vectors are

$$\begin{aligned} f_1 &= \cos\lambda_f \cos\phi_f + \sin\lambda_f \cos d_f \sin\phi_f \\ f_2 &= \cos\lambda_f \sin\phi_f - \sin\lambda_f \cos d_f \cos\phi_f \\ f_3 &= -\sin\lambda_f \sin d_f \\ n_1 &= -\sin\phi_f \sin d_f \\ n_2 &= \cos\phi_f \sin d_f \\ n_3 &= -\cos d_f \end{aligned}$$

Moment Tensor

Jost and Herrmann (1989) related the Green's functions in the formulation of Herrmann and Wang (1985) to a moment tensor source representation. An error entered into the Jost and Herrmann (1989) equations A5.4 - A 5.6, which are correct for a pure deviatoric source and for a pure isotropic source. The correct expressions are as follow:

$$
\begin{aligned}
\mathbf{u_z(r, z = 0, \omega)} = \mathbf{M_{xx}}&\left[\frac{\mathbf{ZSS}}{\mathbf{2}} \mathbf{\cos(2\phi)} - \frac{\mathbf{ZDD}}{\mathbf{6}} + \frac{\mathbf{ZEX}}{\mathbf{3}} \right] \\
&+ \mathrm{M_{yy}}\left[\frac{-\mathrm{ZSS}}{2} \cos(2\phi) - \frac{\mathrm{ZDD}}{6} + \frac{\mathrm{ZEX}}{3} \right] \\
&+ \mathrm{M_{zz}}\left[\frac{\mathrm{ZDD}}{3} + \frac{\mathrm{ZEX}}{3} \right] \\
&+ \mathrm{M_{xy}}\left[\mathrm{ZSS} \sin(2\phi) \right] \\
&+ \mathrm{M_{xz}}\left[\mathrm{ZDS} \cos(\phi) \right] \\
&+ \mathrm{M_{yz}}\left[\mathrm{ZDS} \sin(\phi) \right]
\end{aligned} \tag{2a}
$$

$$
\begin{aligned}
\mathrm{u_r(r, z = 0, \omega)} = \mathrm{M_{xx}}&\left[\frac{\mathrm{RSS}}{2} \cos(2\phi) - \frac{\mathrm{RDD}}{6} + \frac{\mathrm{REX}}{3} \right] \\
&+ \mathrm{M_{yy}}\left[\frac{-\mathrm{RSS}}{2} \cos(2\phi) - \frac{\mathrm{RDD}}{6} + \frac{\mathrm{REX}}{3} \right] \\
&+ \mathrm{M_{zz}}\left[\frac{\mathrm{RDD}}{3} + \frac{\mathrm{REX}}{3} \right] \\
&+ \mathrm{M_{xy}}\left[\mathrm{RSS} \sin(2\phi) \right] \\
&+ \mathrm{M_{xz}}\left[\mathrm{RDS} \cos(\phi) \right] \\
&+ \mathrm{M_{yz}}\left[\mathrm{RDS} \sin(\phi) \right]
\end{aligned} \tag{2b}
$$

$$
\begin{aligned}
\mathrm{u_\phi(r, z = 0, \omega)} = \mathrm{M_{xx}}&\left[\frac{\mathrm{TSS}}{2} \sin(2\phi) \right] \\
&+ \mathrm{M_{yy}}\left[\frac{-\mathrm{TSS}}{2} \sin(2\phi) \right] \\
&+ \mathrm{M_{xy}}\left[-\mathrm{TSS} \cos(2\phi) \right] \\
&+ \mathrm{M_{xz}}\left[\mathrm{TDS} \sin(\phi) \right] \\
&+ \mathrm{M_{yz}}\left[-\mathrm{TDS} \cos(\phi) \right].
\end{aligned} \tag{2c}
$$

Point Force

The displacements corresponding to Green's functions for an arbitrarily point force, given by the vector $\mathbf{f} = (f_1, f_2, f_3)$ are

$$u_z = (f_1 \cos\phi + f_2 \sin\phi)ZHF + f_3 ZVF$$

$$\mathbf{u_r = (f_1 \cos\phi + f_2 \sin\phi)RHF + f_3 RVF}$$

and

$$u_\phi = (f_1 \sin\phi - f_2 \cos\phi)\mathrm{THF}\ ,$$

where the positive displacements in the z, r, and ϕ directions are taken to be *up, away from the source,* and *clockwise about the source when looking down.* The force vector, though, uses a cartesian coordinate system such that the vertical is positive downward, the *1* is north and the *2* is east, and ϕ is measured clockwise from north.

Point Explosion

For an explosion source, the displacements are just

$$u_z = \mathrm{ZEX}$$

and

$$u_r = \mathrm{REX}$$

and

$$u_\phi = 0$$

The pressure field in a fluid due to an explosion is given by the function PEX, which is defined such that a positive value corresponds to a compression.

Units

The physical units of the Green's functions depend upon the source time function and the input earth model. For convenience and historical reasons, the following mixed units are often used:

	Input	Equivalent	
Parameter	User Units	CGS Units	MKS Units
Distance	km	10^5 cm	10^3 m
Velocity	km/s	10^5 cm/s	10^3 m/s
Density	gm/cm^3	1 gm/cm^3	10^3 kg/m^3

If the source time function, s(t), behaves as a step-like function with unit final offset, then its derivative has a unit area. The Fourier transform of the ZEX, REX, PEX and RVF Green's functions for displacement and stress are

$$\mathrm{ZEX} = \frac{M_I S(\omega)}{4\pi\rho\alpha^2}\, e^{\frac{-i\omega R}{\alpha}} \left[\frac{h}{R^3} + \left(\frac{i\omega}{\alpha}\right) \frac{h}{R^2} \right]$$

$$\mathrm{REX} = \frac{M_I S(\omega)}{4\pi\rho\alpha^2}\, e^{\frac{-i\omega R}{\alpha}} \left[\frac{r}{R^3} + \left(\frac{i\omega}{\alpha}\right) \frac{r}{R^2} \right]$$

$$\mathbf{PEX} = \frac{\mathbf{M_I(i\omega)^2 S(\omega)}}{\mathbf{4\pi\alpha^2}} \frac{\mathbf{1}}{\mathbf{R}} \mathbf{e}^{\frac{-\mathbf{i\omega R}}{\alpha}}$$

$$\mathrm{RVF} = -\frac{\mathrm{F_I S(\omega)}}{4\pi\rho\omega^2}\left[\mathrm{e}^{\frac{-\mathrm{i\omega R}}{\alpha}}\left[\left(\frac{-3\mathrm{rh}}{\mathrm{R}^5}\right)+\left(\frac{\mathrm{i}\omega}{\alpha}\right)\left(\frac{-3\mathrm{rh}}{\mathrm{R}^4}\right)+\left(\frac{\mathrm{i}\omega}{\alpha}\right)^2\left(\frac{-\mathrm{rh}}{\mathrm{R}^3}\right)\right]\right.$$
$$\left. - \mathrm{e}^{\frac{-\mathrm{i\omega R}}{\beta}}\left[\left(\frac{-3\mathrm{rh}}{\mathrm{R}^5}\right)+\left(\frac{\mathrm{i}\omega}{\beta}\right)\left(\frac{-3\mathrm{rh}}{\mathrm{R}^4}\right)+\left(\frac{\mathrm{i}\omega}{\beta}\right)^2\left(\frac{-\mathrm{rh}}{\mathrm{R}^3}\right)\right]\right]$$

At large distances and high frequencies, it is seen that

$$\frac{\mathrm{dZEX}}{\mathrm{dt}} \approx \frac{\mathrm{M_I(i\omega)^2 S(\omega)}}{4\pi\rho\alpha^3} \frac{\mathrm{h}}{\mathrm{R}^2} \mathrm{e}^{\frac{-\mathrm{i\omega R}}{\alpha}}$$

$$\frac{\mathrm{dREX}}{\mathrm{dt}} \approx \frac{\mathrm{M_I(i\omega)^2 S(\omega)}}{4\pi\rho\alpha^3} \frac{\mathrm{r}}{\mathrm{R}^2} \mathrm{e}^{\frac{-\mathrm{i\omega R}}{\alpha}}$$

$$\mathrm{PEX} = \frac{\mathrm{M_I(i\omega)^2 S(\omega)}}{4\pi\alpha^2} \frac{1}{\mathrm{R}} \mathrm{e}^{\frac{-\mathrm{i\omega R}}{\alpha}}$$

If the input is the user units, dimensional analysis shows that the time history velocity will be in cm/s for a isotropic moment of 10^{20} dyne-cm. For the same moment, the pressure field will be in units of 10^5 dyne/cm^2, or equivalently in units of 10^4 nt/m^2 (Pa). Similar analysis will indicate that the default force is 10^{15} dynes.

Because of the mixed units, the program Green's functions must be multiplied by the following scaling factors to obtain the desired output units

Input	Scaling Factor	Output	Green's Function
M_{ij} moment tensor (dyne-cm)	10^{-20}	cm	ZDD, RDD, ZDS, RDS, TDS, ZSS, RSS, TSS, ZEX, REX
F_j force (dyne)	10^{-15}	cm	ZVF, RVF, ZHF, RHF, THF
M_I isotropic moment (dyne-cm)	$10^{-20} \cdot 10^4$	Pa	PEX

This assumes that the model distance units are km and the density is gm/cm^3. Derivatives of time series yield output units of cm/s or cm/s/s.
The program **`fmech96`** requires the moment to be in dyne-cm and the force to be in dynes, and properly adjusts the units to CGS for 3-component time histories to yield displacements in *cm*, velocities in *cm/s* and acceleration in *cm/s/s*).

Geographic Displacements

To convert the radial and tangential components of motion to north and east, the following transformation is used:

$$\mathbf{u_z} = \mathbf{u_z}$$
$$u_N = -\cos\theta\, u_r + \sin\theta u_\phi$$
$$u_E = -\sin\theta\, u_r - \cos\theta u_\phi$$

where θ is the back-azimuth from the receiver to the source, measured clockwise from north.

3. Time Functions

The time function is imposed in the last step of computing Green's functions (Figure 1, Chapter 1) through the programs **gpulse96**, **hpulse96**, **spulse96** or **cpulse96**. The programs share a set of command line flags:

-t	Triangular pulse of base 2 L dt
-p	Parabolic Pulse of base 4 L dt
-l L	(default 1)duration control parameter
-o	Ohnaka pulse with parameter alpha
-i	Dirac Delta function
-a alpha	Shape parameter for Ohnaka pulse
-D	Output is ground displacement
-V	Output is ground velocity (default)
-A	Output is ground acceleration
-F rfile	User supplied pulse
-m mult	Multiplier (default 1.0)
-OD	Output is ground displacement
-OV	Output is ground velocity
-OA	Output is ground acceleration
-Z	zero phase triangular/parabolic pulse, else causal

To impose a direct relationship between source moment or force, all pulses, including the user provided *rfile* used with **-F** *rfile*, must have an area equal to *+1.0*. The internal time functions, triangular (**-t**), parabolic (**-p**), Ohnaka (**-o**) or impulse (**-i)** consist of a single positive pulse, with the first three functions representing a low-pass filtered version of the Dirac delta function.

For reasons of numerical stability, the Computer Program in Seismology 3.20 convention is that these functions represent the first derivative of the displacement on a fault, the explosion moment, or point force. Thus the near-field displacement of a dislocation source is a step, the explosion has a step in moment, and the point forces has a step-like time function. The only difference among the pulses, or equivalently the step offset, is the degree of smoothing.

The flags -A, -V, and -D indicate that the resultant time histories are to be ground acceleration, velocity or displacement, respectively for the step-like time history of offset across the fault, increase in moment at the elastic radius of an explosion, or application of

force. These concepts are illustrated in Figure 1, which computes the TSS Green's functions for the wholespace at short (near-field) and large distances (far-field).

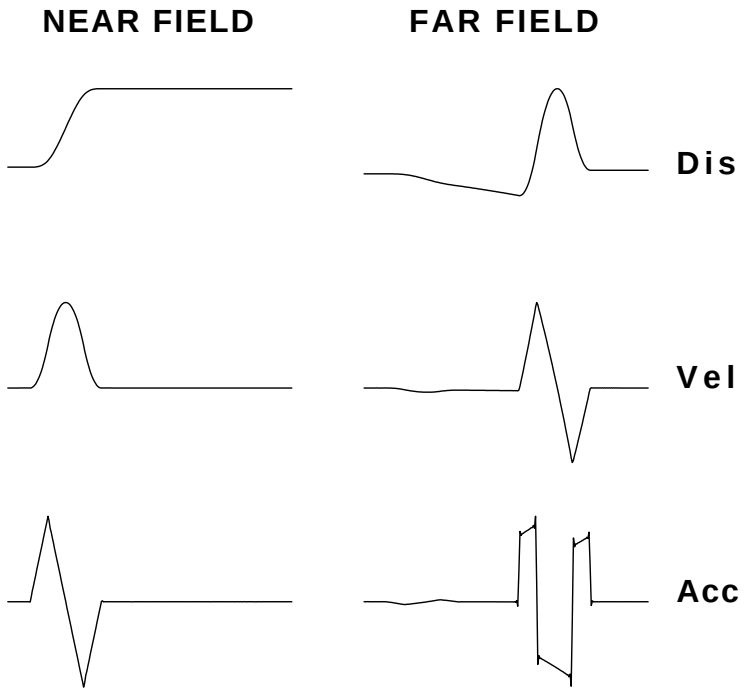


Fig. 1. Comparison of near-field and far-field waveforms for displacement (-D), velocity)-V) and acceleration (-A) command flags for the parabolic pulse.

It is obvious that one way of thinking of the time histories added to the signal using [h,s,g,c]pulse is to associate them with the far-field time function in the seismic wavefield. For example, if you want a displacement seismogram for a step dislocation across the fault, you can apply a delta-function time history to the f96 signals using

hpulse96 -i -D

If you want displacement seismograms for an earthquake that slipped over a time of 4*dt, the far-field time function is a triangle with a duration of 4*dt, then use

hpulse -t -l 2 -D

For clean synthetics, I never use the Ohnaka or impulse sources because these have signal at the Nyquist frequency which causes ringing in the synthetics.

The flags **-A**, **-V**, and **-D** indicate that the resultant time histories are to be ground acceleration, velocity or displacement, respectively for the effective *step* time history at the source. From the discussion above, these flags will give ground acceleration in cm/s^2, ground velocity in *cm/s* or ground displacement in *cm*, respectively, for a default moment of 10^{20} dyne-cm for the ZDD, RDD, ZDS, RDS, TDS, ZSS, RSS, TSS, ZEX and REX Green's functions, and for a default force of 10^{15} dynes for the ZVF, RVF, zHF, RHF and THF Green's functions.

Figures 2 and 3 illustrates the different source pulses upon the near- and far-field synthetics. The triangular and parabolic responses are identical since the *-p -l 1* sampled pulse is identical to the *-t -l* pulse. This was done since the *-t -l 1* would be plagued by

the sharp truncation in the frequency domain.

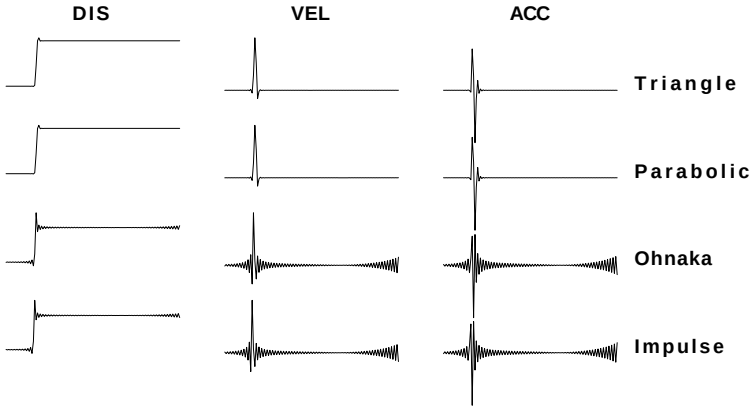


Fig. 2. Comparison of the effect of the pulse type on the displacement, velocity and acceleration waveforms at short distance.

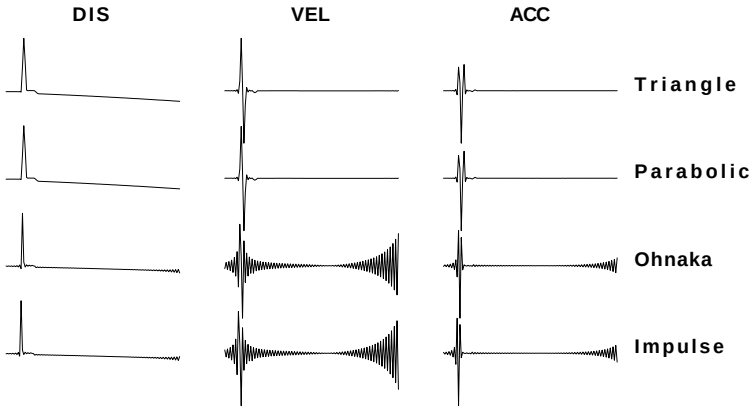


Fig. 3. Comparison of the effect of the pulse type on the displacement, velocity and acceleration waveforms at large distances.

Three other flags, **-OD**, **-OV** and **-OA** have a special purpose for use in getting the true response to an effective impulse source. The Computer Programs in Seismology - 3.20 strongly emphasizes physical units in its **file96** format so that the Green's functions can be used for quantitative source parameter inversion. The emphasis on an effective step-like source time function is appropriate for earthquakes and explosions. To obtain Green's functions for an effective impulse, one just has to take the first derivative of the time series. Unfortunately while **fderiv96** will take the derivative, it will not redefine the **file96** header. The following alternative can be used with any of the **Xpulse96** programs:

To obtain the ground velocity for a parabolic source pulse for an effective step source time function, one invokes

```
spulse96 -p -V
```

If one wants the ground velocity for a parabolic pulse for an *effective impulse source time function*, invoke

```
spulse96 -p -A -OV
```

The **-A** flag forces taking the derivative, the the **-OV** ensures that the header in the **file96** file has *CM/SEC* instead of *CM/SEC/SEC*.

4. Pulses

This section provides more detail on the favored triangular and parabolic pulses:

Evaluation of the inverse Fourier transform to yield a time series is usually accomplished by an inverse Fast Fourier Transform (Brigham, 1974) which approximates the true inverse transform

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{i2\pi ft} d_f$$

by the inverse Discrete Fourier Transform

$$g(k\Delta t) = \sum_{n=0}^{N-1} G(n\Delta f) e^{i2\pi nk\Delta t\Delta f} \Delta f \quad \text{for } k = 0, \ldots, N-1$$

where $\Delta f = \frac{1}{N\Delta t}$. (Note that this is related to the original Fourier transform definition defined in the introduction if $h(t) = g(t)$ and $G(f) = H(2\pi f)$). This approximation to the continuous Fourier Transform introduces concerns about finite frequency and time windows and periodicity in the time and frequency domains. This will always be present when the discrete Fourier transform pair is used, but can be controlled so that the discrete Fourier transform result is a reasonable approximation to the desired Fourier transform solution.

The first problem considered concerns the effect of the finite frequency window in the resultant time histories. The function H(f) is sampled in the range $(-f_N, f_N)$ where $f_N = \frac{1}{2\Delta t}$. Frequencies outside this range are effectively set to zero. If the transitions through the frequency points $f = \pm f_N$ are not smooth, high frequency ripples will be seen in the time domain. This effect can be significantly reduced in synthetic seismograms by choosing source time functions that have zeros at the Nyquist frequencies $\pm f_N$. Two possible functions that have this property are the triangular and parabolic pulses.

Triangular:

$$s(t) = \frac{1}{\tau} \begin{cases} 0 & t \le 0 \\ \frac{t}{\tau} & 0 \le t \le \tau \\ 1 - \frac{t}{\tau} & \tau \le t \le 2\tau \\ 0 & t \ge 2\tau \end{cases}$$

which has the Fourier transform

$$e^{-i\omega\tau}\left[\frac{\sin \pi f \tau}{\pi f \tau}\right]^2$$

This is a positive pulse with unit area and a corner frequency $\mathbf{f_c = 1/\pi\tau}$. If $\tau = \mathbf{2M\Delta t}$ and $\mathbf{M \geq 1}$, this function has spectral zeros at frequencies $\frac{\mathbf{1}}{\mathbf{M}}\mathbf{f_N}$, $\frac{\mathbf{2}}{\mathbf{M}}\mathbf{f_N}$, **. . . ,** $\mathbf{f_N}$, where $\mathbf{f_N}$ is the Nyquist frequency defined as $\mathbf{f_N = \frac{1}{2\Delta t}}$. Thus if we sample the Fourier transform of this pulse and apply the inverse discrete Fourier Transform, we will see a nice sampled triangular pulse without any ripples in the time domain.

Parabolic:

$$s(t) = \frac{1}{2\tau}\begin{cases} 0 & t \leq 0 \\ 1/2(t/\tau)^2 & 0 < t \leq \tau \\ -1/2(t/\tau)^2 + 2(t/\tau) - 1 & \tau \leq t \leq 3\tau \\ 1/2(t/\tau)^2 - 4(t/\tau) + 8 & 3\tau < t \leq 4\tau \\ 0 & t > 4\tau \end{cases}$$

The Fourier transform of this function is

$$e^{-i\omega 2\tau}\,\frac{[2\sin\omega\tau - \sin 2\omega\tau]}{(\omega\tau)^3} \;=\; e^{-i\omega 2\tau}\,\frac{4\sin^2\frac{\omega\tau}{2}\sin\omega\tau}{(\omega\tau)^3}\;.$$

This time function has a unit area and a corner frequency $f_c = 1/4.575\tau$. In addition, it has spectral zeros at certain frequencies. If $\tau = M\Delta t$, where $M > 1$, then spectral zeros are at frequencies $\frac{1}{M}f_N$, $\frac{2}{M}f_N$, . . . , f_N. By choosing τ and Δt such that one of the spectral zeros occurs at the Nyquist frequency, the pulses can be synthesized and propagated through the model without the rippling introduced by an arbitrary, sharp high frequency spectral cutoff. Note that the parabolic pulse with $\tau = \Delta t$ will give the same sample values at the triangular pulse with $\tau = 2\Delta t$.

5. References

Haskell, N. A. (1963). Radiation pattern of Rayleigh waves from a fault of arbitrary dip and direction of motion in a homogeneous medium, *Bull. Seism. Soc. Am.* 53, 619-642.

Haskell, N. A. (1964). Radiation pattern of surface waves from point sources in a multi-layered medium, *Bull. Seism. Soc. Am.* 54, 377-393.

Herrmann, R. B. (1975). A student's guide to the use of P and S wave data for focal mechanism determination, *Earthquake Notes* 46, 29-40.

Herrmann, R. B. and C. Y. Wang (1985). A comparison of synthetic seismograms, *Bull.*

Seism. Soc. Am. 75, 41-56.

Jost, M. L. and R. B. Herrmann (1989). A student's guide to and review of moment tensors, *Seism. Res. Let.* 60, 37-57.

Wang, C. Y. and R. B. Herrmann (1980). A numerical study of P-, SV-, and SH-wave generation in a plane layered medium, *Bull. Seism. Soc. Am.* 70, 1015-1036.

APPENDIX C
PROGRAM LIMITATIONS

1. Introduction

The purpose of this section is to describe limitations in the distributed version of *computer Programs in Seismology*.

2. Volume II

3. Volume III

Program	Parameter	Description
all	parameter(NL=100)	Maximum number of layers in model
sdisp96	parameter(MAXMOD=2000)	Maximum number of modes
sregn96		
slegn96		
sdpsrf96		
sdpegn96		
spulse96	parameter(NSAMP=8192)	Maximum length of time series

4. Volume IV

Program	Parameter	Description

5. Volume V

Program	Parameter	Description
gpulse96	parameter (NSAMP=2048)	Maximum length of time series
gprep96	parameter (NL=50)	Maximum number of layers

		in model
genray96	parameter (NL=50)	Maximum number of layers in model
all f96 programs	parameter (NSAMP=16384)	Maximum length of time series
ffilt96	parameter(NPZ=100)	Maximum number of poles and zeors
shwsac	parameter(LN=20000)	Maximum length of SAC time series
sactoasc	parameter(LN=20000)	Maximum length of SAC time series
asctosac	parameter(LN=20000)	Maximum length of SAC time series
f96tosac	parameter(NSAMP=16384)	Maximum length of SAC time series
sactof96	parameter(NSAMP=8192)	Maximum length of SAC time series

6. Volume VI

Program	Parameter	Description
hpulse96	parameter (NSAMP=4096)	Maximum length of time series
all	parameter (NL=100)	maximum number of layers in model
	parameter(NSOURCE=100)	Maximum number of source depths
	parameter(NRECEIVER=100)	Maximum number of receiver depths
	parameter(NSR=100)	Maximum number of sources and receivers together
	parameter(NDST=100)	Maximum number of receiver distances

7. Earth Models

At present only isotropic, constant velocity layering is supported. Volume III supports a spherical to flat earth transformation that must be examined carefully in the context of its use for making synthetic seismograms.

Volume III permits a stack of fluid layers at the surface, and permits buried sources and receivers in the fluid (as of 1 Nov 96).

Volume V does not permit fluid layers.

Volume VI permits a stack of fluid layers at the surface, and permits buried sources and receivers in the fluid.

Note (as of 1 Nov 96) the earth model specification can permit a definition of the surface for atmospheric problems. consider the following model:

```
MODEL.01
ARDC LINEAR GRADIENT MODEL
ISOTROPIC
KGS
FLAT EARTH
1-D
CONSTANT VELOCITY
LINE08
LINE09
LINE10
LINE11
  H       VP       VS      RHO       QP     QS   ETAP ETAS REFP  REFS
-40      .328     .000    .000220    1000  1.434    0     0    1      1
-40      .338     .000    .001220    1000  1.434    0     0    1      1
40.0     6.10     3.50      2.70     1000    0      0     0    1      1
 0.0     8.10     4.70      3.30     0       0      0     0    1      1
```

The negative layer thickness indicate layering above the surface. The surface is defined at the point between the negative and positive layer thicknesses. To place a source in the atmosphere 10 km above the surface, use HS=-10. To place a source 10 km in the elastic medium, use HS=10. The value of Qs is the specific heat ratio for the atmosphere, which is required is gravity is accounted for.

As of 1 November 1996, gravity has not been built into the codes

APPENDIX D
WORK TO DO NOT UPDATED

1. Introduction

The purpose of this section is to describe necessary modifications and additions to the present code.

- For surface wave code, permit fluid layer stack (done 1 Nov 96), and also permit rigid, free and halfspace boundary conditions at the top and bottom boundaries to be compatible with the wavenumber integration code (still must be done 1 Nov 96)
- For the surface wave code, determine the lowest medium velocity near the source and receiver, and evaluate propagators down and up to this zone. At present propagators go from bottom up. This is required for numerical stability, especially for the case of a layered wholespace, e.g., coal seams
- Build sphericity in to the omega-k code. Also review all previous earth flattening approximations (done 12 Feb 08)
- For fluid sources, consider the implications of the energy injection source and also replacing the atmospheric explosion by a point vertical force at the surface. This is mentioned in Ben-Menahem and Singh, and is valid for solid elastic waves since the the phase velocities of seismic arrivals, P, S and Rayleigh will have sqrt(k**2 - ka**2) approximately ka, e.g., you would get this by stationary phase.
- Build in transverse anisotropy (done for wavenumber integration 10 JUN 2002)

APPENDIX E
INSTALLATION REVISED

1. Introduction

The purpose of this section is to describe the installation of these programs.

• Install the programs using

gunzip -c NP330.tgz | tar xvf -

or

This will create the directory PROGRAMS.330

cd PROGRAMS.330

Now run the script *Setup* to learn the options:

```
Checking for existence of bin and lib directories
Directory bin exists
Directory lib exists
Usage: Setup SOL WIN32 SOL-EGCS LINUX LINUX64 CYGWIN OSX
  LINUX6440  Linux 64 bit with gcc/gfortran compilers
  OSX40      Apple  with 64-bit gcc/gfortran compilers
 ----- deprecated since most computers are now 64bit e -----
  LINUX40    Linux with gcc/gfortran compilers
  OSX40-32   Apple  with 32-bit gcc/gfortran compilers
 ----- not tested in a long while -----
  CYGWIN     CYGWIN 98/NT/2K/XP gcc/g77 Compilers
  CYGWIN40   CYGWIN 98/NT/2K/XP gcc/gfortran Compilers
```

Now run this as

This says the target is a LINUX system with a 64-bit processor, which is standard today. All output of the run is in the *C.txt* file. The command line assumes that the **sh** or **bash** shel is being used. If there are any problem with the compile, these will be documented in this file.

The last few lines of this file will look like

Compile of Computer Programs in Seismology is Over
158 programs were compiled

Finally set your PATH variable in your environment include the path to the CPS bin directory. For example if CPS is in my ${HOME}/PROGRAMS.310t/PROGRAMS.330/bin, then I do this in the *.profile* file in my login directory:

PATH=$PATH
PATH=:.:$HOME/bin:$HOME/PROGRAMS.310t/PROGRAMS.330/bin:$OPATH