VOLPIS CODE - A SHORT GUIDE FOR USERS

Version 9, S. Cesca, University of Hamburg, November 2006

INTRODUCTION

VOLPIS is a Fortran 77 code to invert source parameters of volcanic seismic events. The code requires the previous processing of data, generation of Green´s functions (GF) and selection of numerical parameters. The output is given in terms of 9 time-dependent source components: 6 components of Moment Tensor (MT), 3 components of Single Force (SF).

This guide will not enter in the details of code structure and methodology, for which a large description is given in the research paper. In the following, it will be explain the format of input/output data and the basic informations needed to run the code in a linux system.

The original version of the code, as well as additional files containing this short guide, scripts and examples, are submitted to "Computers and Geosciences" journal together with the submitted research paper: Cesca, S. and Dahm, T., 2006. "A frequency domain inversion code to retrieve time-dependent parameters of long period volcanic sources".

For further information, comments or reports of any bugs, please contact:

Dr. Simone Cesca
Institut für Geophysik, University of Hamburg
simone.cesca@zmaw.de

(now at SZGRF Erlangen, BGR; simone.cesca@zmaw.de)

VERSION 9 (AND VERSION 9b)

Version 9 (volpis9.exe) is the last released version of this code. It was coded in the end of 2006. It includes all features from previous versions, such has the handling of different data formats and the application of source constraints by means of a singular value decomposition.
Additionally, version 9 include an error estimation for source components. This additional result is achieved by using a bootstrap approach, which test 200 perturbed solutions.

In view of the portability of the code to other operative systems, and to insure a correct compilation when using different compilers than g77, a second version (named volpis9b) has been included. This version is in all equivalent to volpis9, except that SAC input is not included (and therefore precompiled SAC library for linux are not required).

GETTING STARTED

1) Download the package (volpis9.tar.gz)
2) Uncompress and extract the directory structure (use: gunzip, tar -xvf)
3) Compile the code versions (make volpis9, make volpis9b, respectively for the two versions)
4) Move to your working directory all needed files: Greens functions, data, data window input file (such as retard.dat) and executable (volpis9.exe)
5) Create your input file as volpis9.input (see below for details relative to the input file parameters)
6) Execute the code: volpis9.exe <volpis9.input (or, volpis9b.exe <volpis9.input)

INPUT FILES

The code execution requires the previous preparation of different files:
1) data files (displ.c.n),
2) GF files (MTx.c.n, SFy.c.n),
3) time windows file (retard.dat),
4) numerical parameter file (volpis9.input)
These files are descripted in the following.

1. Data files

Data files include time traces for data. They can either correspond to displacements, velocity of acceleration, and can be referred to any of the following axes: radial (r), transversal (t), vertical (z), north (n) and east (e).

For each station, there should be a different file. Each station has to be numbered, and this numbering should be kept unchanged in the whole process.

Data files have to be saved with the following naming:

displ.c.n, where c stands for component and n for station number (ex. displ.z.4 will contain vertical component of station 4)

Data can be saved either in SAC binary format (not valid for volpis9b) or an ascii-compressed format related to reflectivity method (see Appendix B).

Concerning SAC format, the following parameters of sac header have to be set:

Set CMPAZ, CMPINC, STLO, STLA, EVLO, EVLA, DELTA and NPTS to proper values.
Set DELTA to the same value for all traces
Set B to 0
Set USER1 additional variable as:

1: radial component
2: vertical component
3: transversal component
7: north component
8: east component

2. Green's Functions files

GF files include responses of simple excitations (etiher any of MT or SF components). They consist of time traces. The following rules has to be followed:

Time traces: use displacements, velocities or accelarations, in agreement with data.
Components: use the same reference axes as for data;
Numbering of stations: same numbering as for data;
Sampling rate: same as for data; the same for each trace;
File naming: use names MTx.c.n for responses of Moment Tensor source components (x number of source component, 1..6, see the research paper for conventions; c, component; n, station number) and SFy.c.n for responses to single forces (y number of SF component, 1..3, see the research paper for conventions; c, component; n, station number);

File format: either SAC binary format (not for volpis9b) or an ascii-compressed format related to reflectivity method (see Appendix A).

SAC binary files: parameters of SAC header have to be set as for data.

3. Time windows file

This file is named retard.dat.
First line is ./
All other lines (one for each trace time data window to be used for the inversion) have the following format:

column 1 component (R,T,Z,N,E). Use capital letters;
column 2-5 station number (in agreement with data and GF);
column 7-10 station name (character, maximal length 4 characters);
column 14 P (p-wave) or S (s-wave)
column 18-22 (unused)
column 25-28 beginning of time window for data (in number of points)
column 31-34 beginning of time window for GF (in number of points)
column 37-43 unused
column 46-52 station azimuth (in degrees)
column 54-57 length of data windows (in number of points)
column 59-63 Weight of time trace data window

Note: if running the inversion, lengths of data windows should be constant, if running the forward model option, they can be different.

4. Numerical parameter file

This file is named volpis9.input
Each line contain information for the numerical parameter of code execution:

Line 1:
- name of time windows file (e.g., retard.dat)

Line 2:
- number of points for FFT calculation (power of 2, larger than data window length),
- percentage of time window to be used for taper in time domain (negative values enlarge the original time-window size on both sides)
- lower corner value for frequency bandpass filter
- higher corner value for frequency bandpass filter
- distance dependent weight (0=no scaling, 1=scaling proportional to r)
- 2 (do not change)
- 1 (do not change)

Line 3:
- unused. Keep unchanged (1,1,99999,99999)

Line 4:
- Input format (0=13bit ascii format, 2=SAC binary)
- Output format (0=13bit ascii format, 1=GSE format)

Line 5:
- Execution mode swith (0=inversion, 1=forward modeling)
- Length of STF (Fz. Brustle-Muller) to be used for the forward modeling; set to 0 if running inversion, or if STF is already included in GF.

- Switch for forward modeling source type (if running inversion set to 0).
  If running the forward modeling, accepted values are:
    0: general source (MT+SF)
    1: pure crack
    2: pure double couple (DC)

If running the forward modeling additional lines have to be included, depending on the selected source type:

- If source type 0 (general source):
    Line 6: M11, M12, M22, M13, M23, M33
    Line 7: F1, F2, F3

- If source type 1 (pure crack):
    Line 6: Strike angle, Dip angle, Mu, Lambda
    (angles given in degrees, Mu and Lambda are Lame´ parameters)

- source type 2 (pure double couple):
    Line 6: Strike angle, Dip angle, Rake angle
    (angles given in degrees)

EXECUTION

At the moment of execution all input files have to be saved in the working directory.
The code can be then executed by the command:

volpis9.exe <volpis9.input (or, volpis9b.exe <volpis9.input)

Use the script testlocation.csh to run the inversion for a set of possible Green´s functions (e.g., to test different locations). This need data files to be saved in the subdirectory DATA, each set of GF to be set in subdirectories named GREENn (n=number of GF set), and retard.dat files also saved in the same subdirectories GREENn.

ERRORS AND WARNINGS DURING EXECUTION

In the following, we briefly describe the possible problems arising during the execution, by explaining the meaning of error and warning messages, and giving helpful hints on how to solve the problems.

The following ERROR messages may appear during execution. Error messages are accompanied by the interruption of the execution:

ERROR: Source type, incorrect input
> In the numerical parameters input file (volpis9.input) a wrong value has been chosen for the source type (last value of line 5, if running a forward modeling)

ERROR: incorrect source type
> In the numerical parameters input file (volpis9.input) a wrong value has been chosen for the source type (last value of line 5, if running a forward modeling)

reading error , ierr=number
> Error in reading SAC formatted files (may occur both for data or GFs). Check SAC manual for more indications on error type.

ERROR!!! kbeg < 0

> An error occur during tapering of selected data windows. This may happen if the data window start too soon (check values at columns 25-28 in retard.dat) and/or a too large external taper was chosen (check second value in line 2 in volpis9.input)

Error!!! i2 < 0, stop

> An error occur during selection of GF windows. This may happen if the data window start too soon (check values at columns 31-34 in retard.dat) and/or a too large external taper was chosen (check second value in line 2 in volpis9.input)

Error : igame incorrect

> Set correctly last two values in line 2 in volpis9.input (they have to be 2 and 1)

ERROR: Inversion incorrect

Complex matrices at all frequencies can not be inverted

> Inversion can not be done, since the problem was ill-posed for all considered sampled frequency and none of the complex matrices could not be inverted.

ERROR: isampf must be power of 2

> First value in line 2 in volpis9.input has to be a power of 2.

ERROR: isampf too small, or isamp too large

> First value in line 2 in volpis9.input (here referred as isampf) has to be larger than window length (isamp). Note that both values are expressed in number of points). You may either increase isampf or reduce window length.

ERROR: fc must be > 0

> Higher corner frequency for the bandpass has to be a positive number. This value is defined by user as the fourth value in line 2 in volpis9.input.

The following WARNING messages may appear during execution:

warning
i=<NUM1> azi .ne. az, r= ....
...corrected

> Azimuth inconsistency between data file and the one given in retard.dat. The problem occurred for the trace number (NUM1). The problem is solved by assuming azimuth as given in retard.dat.

WARNING: stf too long or data too short at

> The chosen duration of the source time function (STF) is excessively long with respect to the length of data windows. A shorter value is preferable. Computation continue, but results should be checked carefully.

WARNING: dt.ne.dtr in <FILENAME>

> Data samling factor in GF (file FILENAME) is different from the one of data. Data sampling from data is assumed to be the correct one.

WARNING: r= <NUM1> .ne.re(j)= <NUM2> in <FILENAME>

> Inconsistency between epicentral distance in GF (NUM1) and in data (NUM2) for GF file (NAMEFILE). Epicentral distance from data is assumed to be the correct one.

OUTPUT FILES

Several output files are given as result of execution:

dispf.c: filtered data (c stays for component; file names are dispf.c.gse, if GSE output is chosen)

synth.c: synthetic data for an unconstrained source (c stays for component; file names have are synth.c.gse, if GSE output is chosen)

syntc.c: synthetic data (c stays for component; file names are syntc.c.gse, if GSE output is chosen) for a source constrained to have a common time history for all components (constrain c.1)

synt2.c: synthetic data (c stays for component; file names are synt2.c.gse, if GSE output is chosen) for a source constrained to have a common time history for MT components and another for SF components (constrain c.2)
dis.c.n: additional output (read by the GMT script) for filtered displacements in 2-columns format (column 1 = time, column 2 = displacements); c stays for component, n stays for station number.
syn.c.n: additional output (read by the GMT script) for synthetic for the in unconstrained source. A 2-columns format (column 1 = time, column 2 = displacements) is used; c stays for component, n stays for station number.
sy2.c.n: additional output (read by the GMT script) for synthetic for the in constrained (c.2) source. A 2-columns format (column 1 = time, column 2 = displacements) is used; c stays for component, n stays for station number.
syc.c.n: additional output (read by the GMT script) for synthetic for the in constrained (c.1) source. A 2-columns format (column 1 = time, column 2 = displacements) is used; c stays for component, n stays for station number.
imtx.dat: ascii file, x source component of MT for the unconstrained source
isfy.dat: ascii file, y source component of SF for the unconstrained source
cmtx.dat: ascii file, x source component of MT for the constrained (c.1) source
csfy.dat: ascii file, y source component of SF for the constrained (c.1) source
mmtx.dat: ascii file, x source component of MT for the constrained (c.2) source
fsfy.dat: ascii file, y source component of SF for the constrained (c.2) source
ymtx.dat,zmtx.dat: ascii files, equal to imtx.dat +/- variance (only in version 9)
ysfx.dat,zsfx.dat: ascii files, equal to imtx.dat +/- variance (only in version 9)
result.fit: contains the fit of data:

- fit of synthetic to data for solution unconstrained
- fit of synthetic to data for solution constrained (c.1)
- fit of synthetic to data for solution constrained (c.2)
- fit of constrained (c.1) solution to the unconstrained one
- fit of constrained (c.2) solution to the unconstrained one

volpis.erg: control file

CONTENT OF DIRECTORY STRUCTURE

./ main source file (volpis9.f, volpis9b.f, etc.)
file with variable declaration (volpis.inc)
compilation command (makefile)
./src source files of needed subroutines
./doc documentation
./examples
example of application. It includes all input and output files as in test A and B described in Cesca and Dahm (2006, submitted to Computers and Geosciences). Some details are given in Appendix A.

Appendix A. EXAMPLES

Two examples have been included with the code. A complete description of the examples parameters is given in Cesca and Dahm (2006, submitted to Computers and Geosciences).

Example A (./examples/TESTA) reproduces a synthetic case for Stromboli volcano.
Greens functions have been obtained using a pseudospectral approach for a 3D model including topography (while the velocity model is homogeneous).

Synthetic data have been generated for an isotropic source (as the sum of three dipoles: M11, M22, M33) of 2s duration starting at t = 1s. The explosion is followed (at t = 3s) by the application of a vertical force of 1s duration. White noise have been superposed to data.
Data and Green functions have been saved in sac format and can be easily visualized using SAC.
Results can be plotted using a GMT script (source lpsource.gmt), which generate a postscript file (lpsource.ps). The plot shows on the left side examples of time traces: filtered data (dashed lines) and synthetics (continuous lines) for the unconstrained source (thick lines), the constrained c.2 source (medium lines) and the constrained c.1 source (thin lines). On the right side it shows the retrieved source components (6 moment tensor components and 3 single forces components) for the unconstrained case (left), the constrained c.2 source (centre) and the constrained c.1 source (right). Dashed lines give error estimation.

Example B (./examples/TESTB) reproduces a synthetic case for Kilauea caldera, Hawaii.
Greens functions have been obtained using a reflectivity approach for a 1D model not including topography, but assuming an horizontally layered structure.
Synthetic data have been generated for a source represented by two horizontal dipoles (M11, M22) and an oblique force (F1=F2=F3), all occurring at t = 0s. White noise have been superposed to data.
Data and Green functions have been saved in an ascii format, which is explained in appendix B.
Results can be plotted using a GMT script (source lpsource.gmt), which generate a postscript file (lpsource.ps). The plot shows the retrieved source components (6 moment tensor components and 3 single forces components) for the unconstrained case (black lines). Red lines give error estimation.

Appendix B. REFLECTIVITY ASCII FORMAT

This ascii format, which is needed for data and Greens functions if running version 9b of the volpis code is here briefly explained:

Line 1: Text, indicating the file content
Line 2: 1, 0, 0, 2, comment
(these values can be kept fixed for execution volpis; in any case the first integer indicates in which layer the source is embedded)
Line 3: 0, 0, 3.84, 1.88, 2.65
(these values can be kept fixed for execution volpis; in any case, the last three values give P wave velocity, S wave velocity and density at the source)
Line 4: Comment
Line 5: Epicentral distance (in km)
Line 6: Reduction velocity; time offset; time sample of data trace (in seconds); 0; 0; text
Line 7: Epicentral distance (in km); 0; component
(second values should be kept at 0)
(third value must indicate the trace component: 1=vertical, 2=radial, 3=tangential, 7=north, 8=east)
Line 8: Number of points of the time trace; Maximal amplitude (the magnitude have to be consistent between data and GFs; Azimuth (in degrees)
Next lines: Data (each line have 16 values represented by a 5 digit integer).
If we name “xint” this integer value and we named “MaxAmpl” the maximal amplitude, then the absolute data value “x” is given by the expression:

$$x = ((xint-49999)/50000) * MaxAmpl$$

Last line: Epicentral distance (in km)

As example of files with this format, you can check data and GFs files of example B.