Instruction

- Analysis of SPAC Method -

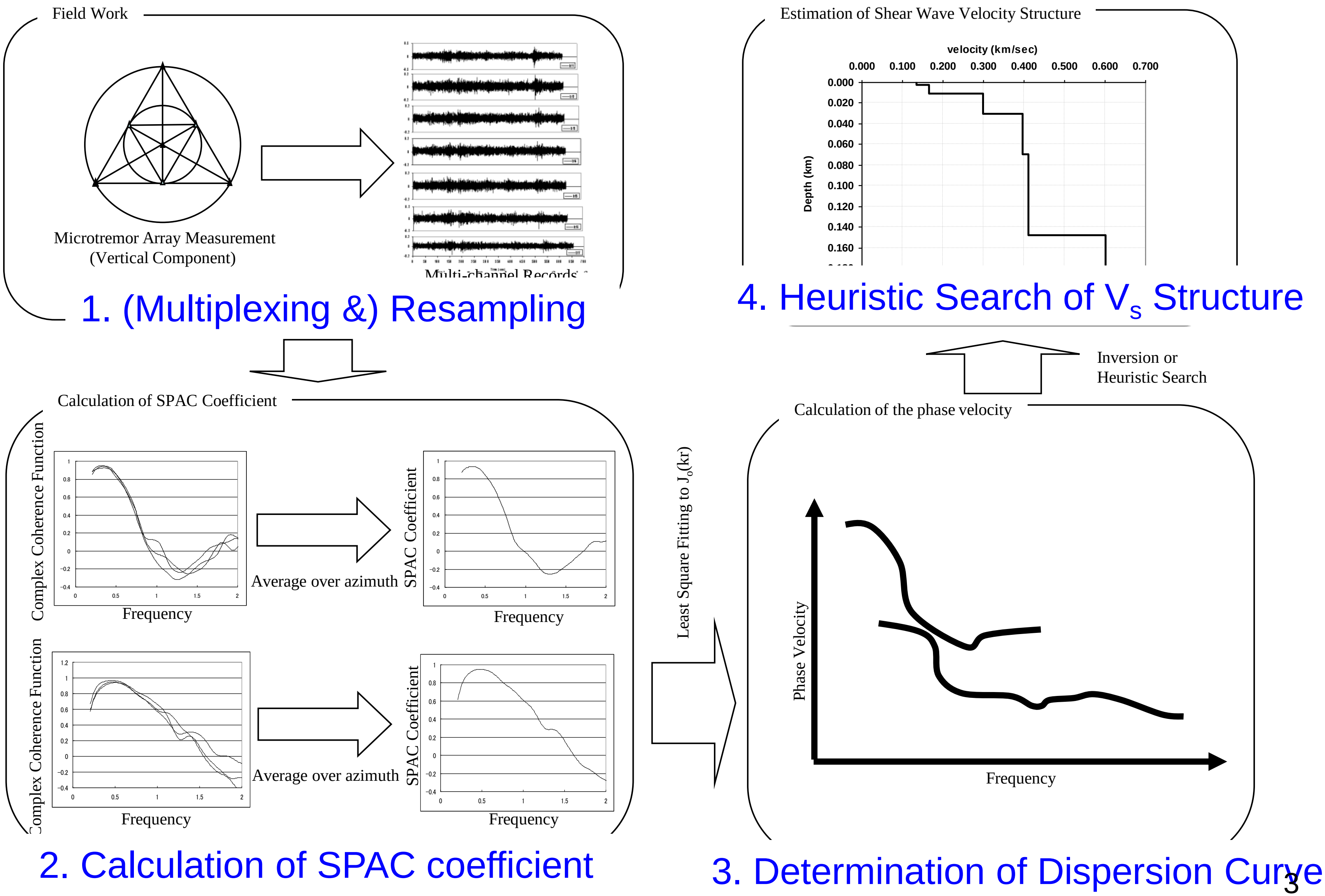
Feb. 10, 2019

IISEE, BRI, Japan

By T. Yokoi

Acknowledgements:
The revision for the 2019 version was partly supported by Japan Science and Technology Agency (JST) and Japan International Cooperation Agency (JICA) under the “Science and Technology Research Partnership for Sustainable Development (SATREPS): Integrated Research on Great Earthquakes andDisaster Mitigation in Nepal Himalaya (FY2016-2020)”.

Procedures of analysis



Note: This version was developed on Linux: Ubuntu 18.04 LTS on VMWare Workstation Player 12.1.1 (build-3770994) on Windows10 Home 64bit (Build 14393) for 64bit PC, using gfortran compiler.

Operation on other OS may require additional revision or modification by users themselves.

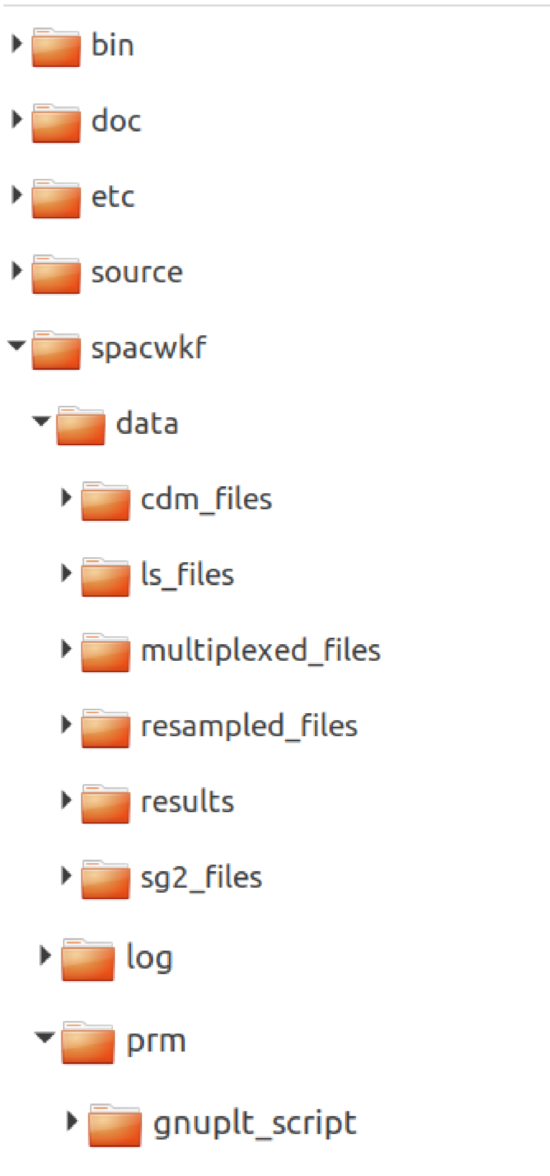
Execution of commands is conducted as

./bin/*executable_file_name*.exe
or
sh *shell_script_file_name*.sh

If it is necessary to leave log file of execution

./bin/*executable_file_name*.exe 2>&1 | tee ./spacwkf/log/*log_file_name*.log
or
sh *shell_script_file_name*.sh 2>&1 | tee ./spacwkf/data/log/*log_file_name*.log

Folder Structure



Every necessary programs and files are stored under the folder “SPAC2019”. The command operation must be conducted in the same folder, where shell script files are stored.

The source codes of the programs are stored in the subfolder “source”, document files including this instruction manual in “doc”, executable ones in “bin”.

The subfolder of work space “spacwkf” contains the subfolder “prm” for parameter files that includes script files of GNUPLOT and the subfolder “data” for data files including graphic ones.

The compressed file “spacwkf.tar.gz” keeps subfolder structure of “spacwkf” and all parameter files in “spacwkf/prm”.

Note: GNUPLOT scripts files

Some files of GNUPLOT scripts are stored under the subfolder

"./spacwkf/prm/gnuplt_scripts"

These can be loaded on GNUPLOT as load '????'
Some programs create the scripts of GNUPLOT that include the command

'set terminal x11' ,

This works on the GNUPLOT on Ubuntu and may be that on Windows.

If any problem on Windows, it is worth to try to replace it with

'set terminal wxt' .

Note: Executable files

The folder “SPAC2019” includes several executable files. Their source code files are stored in the subfolder “./source”. Then, the following command is required to re-compile them if necessary. In the folder SPAC2019, type in the following command.

```
gfortran ./source/???.for  –o ./bin/???.exe
```

In case of problems caused by the incompatibility between Fortran77 and Fortran95,

```
gfortran  -ff2c ./source/???.for  –o ./bin/???.exe
```

Executable files are stored in the subfolder “SPAC2019/bin”.

Note: Shell script files

The folder “SPAC2019” includes several shell script files.
They are composed of few executing commands to reduce the typing tasks in data processing.
The following command can execute the shell script files.

sh *shell_script_file_name*.sh

As the contents of the shell script files contained in this program package are simple, they can work as batch files. However, it is necessary to activate batch files using the following.

chmod u+x *shell_script_file_name*.sh

For execution as a batch file,
shell_script_file_name.sh

Note: Cleaning up the subfolders

- Use
  sh clean_all.sh
  in the folder "SPAC2019" to delete all files of input data, interim outputs and results for a new processing task.
- Use
  sh clean_sg2.sh
  in the folder "SPAC2019" to delete all input files of sg2 format under "./spacwkf/data/sg2_files".
- Use
  sh clean_ls.sh
  in the folder "SPAC2019" to delete all input files of win format under "./spacwkf/data/ls_files", but subfolder structure is kept.

Note: Format of Data Files

- Basically, users themselves are responsible to convert the format of the input data files to one of those acceptable formats by this program package.
- The acceptable format is “*.cdm” described in the section “1. Resampling”.
- The following two format converters are provided:

- sg2 (seg2 standard of IEEE)
  ➔./spacwkf/multiplexed_files/*.dat

- ls (Win format of Hakusan-Kogyo)
  ➔./spacwkf/cdm_files/*.cdm

0. Format conversion

0.1. *seg2* standard format

Shell Script used:

sh seg2read.sh

Program and parameter file used:

seg2read.exe +./spacwkf/prm/seg2read.prm

seg2read.exe is prepared for the field data files of seg2 standard format.

Terminology

Multiplexing:

To sort the data individually stored in single channel files into a multi-channel file of the time-sequential format.

Here, *seg2* standard (multiplexed binary) format (IEEE) is explained.

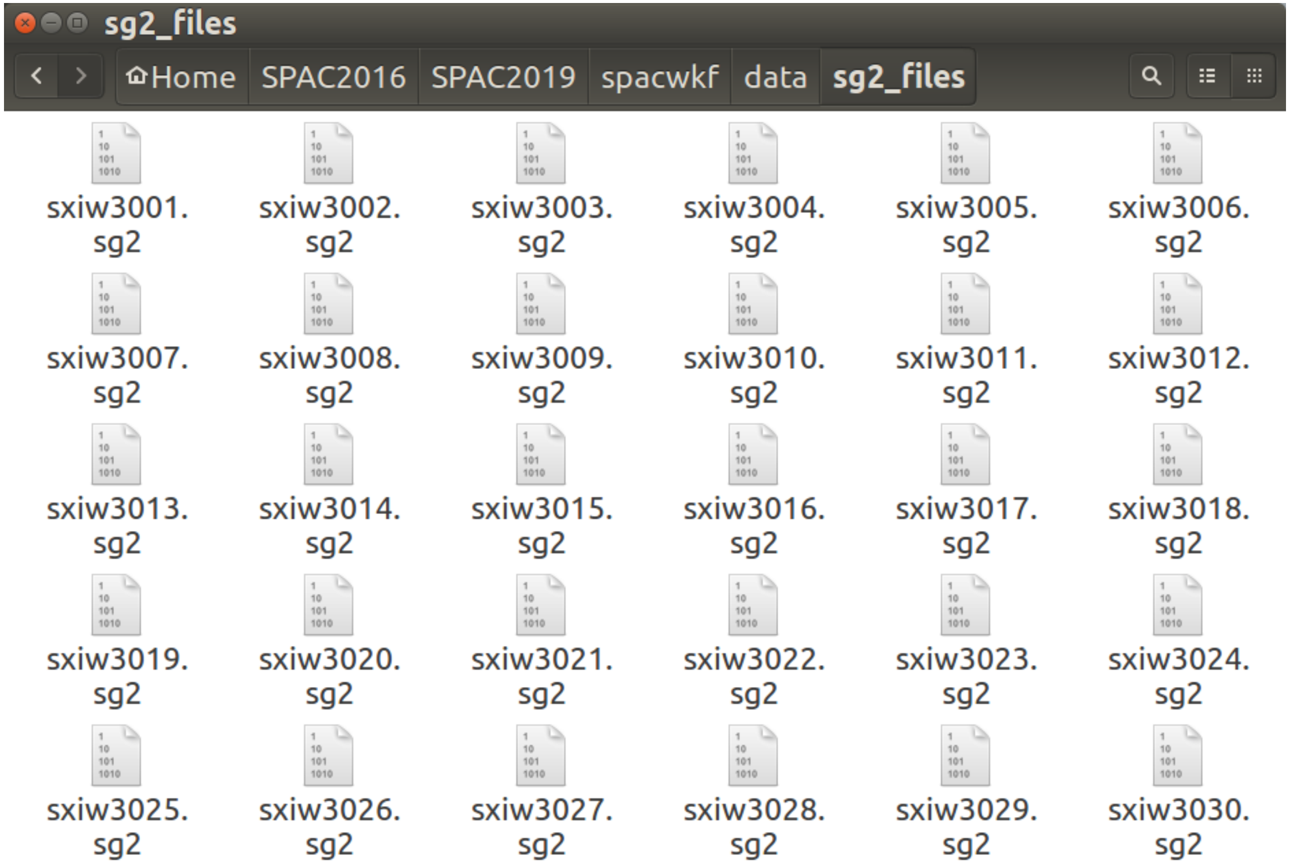
The expected input files of seg2 format are, for example, those obtained in the field using multi-channel data logger designed for the exploration geophysics.

seg2read.exe

+ Convert data format from seg2 standard format (binary & multi-channel) in the sub-folder “spacwkf/data/sg2_files" to cdm format (ascii text, multi-channel),
+ Channel pivoting and extraction
+ Store the output files into “spacwkf/data/multiplexed_files”

First: Copy all the seg2 format files to be converted into the sub-folder “spacwkf/data/sg2_files".

Example:



seg2read.sh:

```
#! /bin/sh -x
cd spacwkf/data/sg2_files
ls *.sg2 > sg2file.lst
cd ../../..
./bin/seg2read.exe | tee spacwkf/log/seg2read.log
cd spacwkf/data/multiplexed_files
ls *.dat > mltfile.lst
cd ../../..
./bin/mk_title.exe
```

Shell script executes "ls *.sg2 > sg2file.lst" in this sub-folder and existing sg2 files are listed in the newly created file "sg2file.lst".
All the files listed in it that have the extension specified in the 3rd line of the parameter file “seg2read.prm”.

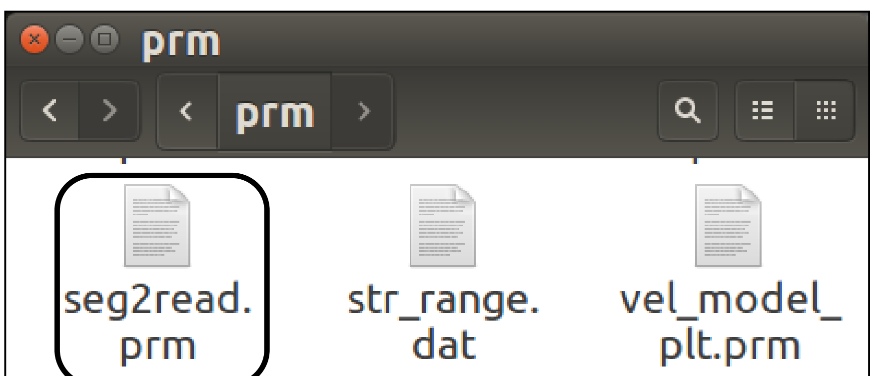
Finally, the first line of “seg2read.prm” is copied to “graph_title.txt” in the subfolder “./spacwkf/prm”.

Example: seg2read.prm

All the files listed in "sg2file.lst" that have the extension specified in the 3rd line are converted to the output files that have the extension given in the 4th line. Edit the file "sg2file.lst" using “gedit” or other text editor if necessary.

seg2read.prm

```
Iwaki City Hall, radius=25m, McSeis/SW, L22 X 7 points    Dec.23 2012 :comment(a7
 0.055        :(A12) scaling factor (for output files in mkine(1.e-3cm/s)/gal(1.e c
sg2           : extension of input seg2 format files(a3)
dat           : extension of output ascii text files(a3)
0 3 0.1 1.0 1.5  :nfilter(=1:apply),nchara=3:bandpass),fl,fh,fs
pvlist 7 6 5 4 3 2 1
```



Explanation: seg2read.prm

All the files listed in "sg2file.lst" that have the extension specified in the 3rd line are converted to the output files that have the extension given in the 4th line. Edit the file "sg2file.lst" using “gedit” or other text editor if necessary.

seg2read.prm

```
1st line : comment (a70)
2nd line : scaling factor (use the value that makes the unit of the output files
           "mkine" (1.e-3 cm/s))
3rd line : extension of input seg2 format files(a3)
4th line : extension of output ascii text files(a3)➔Fix it “.dat”
5th line : nfilter(=0:pass, =1:apply, =2:DC & Trend removal),
           ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
6th line : Channel Pivoting
           'normal' : no pivoting, all channel used
           'rev_al' : all channel used but in reversed order
           'rev_fh' : all channel used but former half in reversed order
           'rev_lh' : all channel used but latter half in reversed order
           ‘pvlist 2 1 3 4 6 23 24’ : Pivoting list.
```

Examples of the 6th line of seg2read.prm

```
Use all channels without pivoting:
       normal    : Channel Pivoting
Use all channels but reversed order:
       reverse   : Channel Pivoting
Use the first 7 channels of the input files without changing order:
       pvlist 1 2 3 4 5 6 7 : Channel Pivoting
The same as above but 7th channel moved to the first:
       pvlist 7 1 2 3 4 5 6 : Channel Pivoting
Use only odd numbered channels among 24 without changing order:
       pvlist 1 3 5 7 9 11 13 15 17 19 21 23 : Channel Pivoting
```

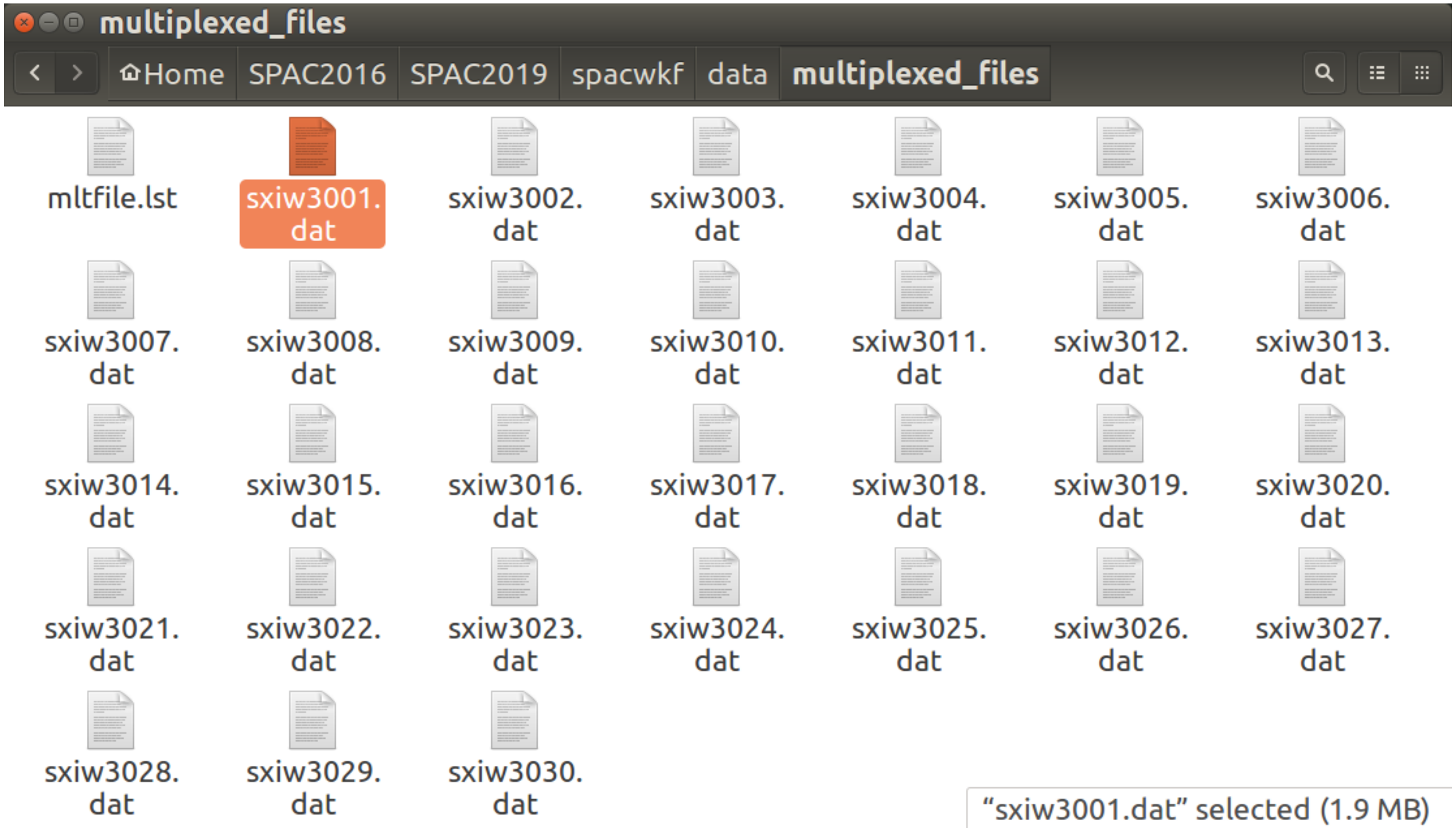
Note: Be sure to put ‘ ‘(blank) before ‘:’(colon), otherwise the program can have an error in detecting the end of line.

Execution

```
 File  Edit  View  Search  Terminal  Help
yokoi@ubuntu:~/SPAC2016/SPAC2019$ sh seg2read.sh
./spacwkf/prm/seg2read.prm
   5.49999997E-02 mkine
./spacwkf/data/sg2_files/sg2file.lst
 sxiw3001.sg2

 sxiw3002.sg2

 sxiw3003.sg2

 sxiw3004.sg2

 sxiw3026.sg2

 sxiw3027.sg2

 sxiw3028.sg2

 sxiw3029.sg2

 sxiw3030.sg2

    30 files have been converted.
Normal End.
```

Input files are copied in ‘./spacwkf/data/sg2_files’.
Multiplexed outputs are stored &./spacwkf/data/multiplexed_files”.
Log file is stored in ‘spacwkf/log/seg2read.log’

The converted files are stored in “./spacwkf/data/multiplexed_files”



Note: The converted files are already multiplexed and stored in the subfolder “./spacwkf/data/multiplexed_files” with the extension “.dat”. For these files, the next step is resampling.

Format of output files in ./spacwkf/data/multiplexed _files:

Users who use single channel recorders or data loggers must multiplex the record files in the following format by themselves.

```
 7  0.0020      0.5500E-01   16384  mkine
Iwaki City Hall, radius=25m, McSeis/SW, L22 X 7 points    Dec.23 2012
    0.000000 -0.3943280E-05  0.9517429E-05 -0.2661598E-06 -0.7497172E-05  0.34044
    0.002000 -0.4349669E-05  0.9327264E-05  0.5046593E-06 -0.8176097E-05  0.35241
    0.004000 -0.4659054E-05  0.9203497E-05  0.5160243E-06 -0.8879738E-05  0.35992
    0.006000 -0.4855397E-05  0.9563433E-05 -0.1432124E-06 -0.9843301E-05  0.37105
    0.008000 -0.6745830E-05  0.1001426E-04 -0.5579629E-06 -0.1409490E-04  0.37764
```

1st line: Number of channels, Δt(sec), scale, number of samples,unit (mkine or gal)
2nd line: Comment (less than 50 characters)
3rd line: Time, 1st-ch sample, 2nd-ch sample, 3rd-ch sample, ….

In the next step (resamplec.for reads this file as follows)

```
read(1,*)nch00,dt00,scale00,ndata00,cunit
…
read(1,'(a50)')comment
…
read(1,*,end=10) xdum,(x(i,j),j=1,nch)
```

Warning!

seg2read.exe can handle less than or equal to 25 channels and less than or equal to 500,000 samples in every channel.
Exceedance may result in a significant error.
It is recommendable to split the input data file if too long, for example, into several files of 1 hour or 30 minutes data.

Ref:
500,000 samples correspond to 1.38 hours for 100 Hz sampling,
1.11 hours for 125 Hz sampling,
41.6 minutes for 200 Hz sampling,
33.3 minutes for 250 Hz sampling
16.6 minutes for 500 Hz sampling.

Output files in "./spacwkf/data/multiplexed _files"

Jump to "1.2. Plot Waveform".

0. Format conversion

0.2. *win* format for LS8800

This is the example of individually recorded data using a tri-axial sensor and three channel data logger at each observation point.
Format conversion & Multiplexing must be done by the users prior to the analysis for the case of individual recording at each site.
Here *win* format is explained. The *win* format data files are created, *e.g.*, LS8800 of Hakusan Kogyo.
The converted files must be written in a format that is readable in the next step: multiplexing.
As it is impossible to cover all existing formats in the world, it is strongly recommended for users to make their own program for format conversion.

Format conversion is conducted using
sh lstocdm2.sh
in the folder “SPAC2019”.

Note: Usage of 4 seismographs in a site is assumed.

Preparation:
1) Edit the parameter file “prm_maker.prm”

```
Sitename_   : site name (a9)
 3  3       : numbers of obs_ponts and channels
 10          : duration of each connected file in min.(integer)
17091511.45 : first file name (yymmddhh.mm)
 20         : number of output connected files (integer)
```

2) Copy the data files (binary) of LS8800 into the subfolders of
“spacwkf/data/ls_files” as follows:
“no1”🡐 files from seismograph No.1
“no2”🡐 files from seismograph No.2
“no3”🡐 files from seismograph No.3
“no4”🡐 files from seismograph No.4

lstocdm2.sh

```
#!/bin/sh
./bin/prm_maker.exe | tee spacwkf/log/prm_maker.log
./bin/lstocdm2.exe | tee spacwkf/log/lstocdm2.log
cd ./spacwkf/data/cdm_files
rm *.cdm
cd ../../..
mv ./spacwkf/data/ls_files/Combined_Data/*.cdm ./spacwkf/data/cdm_files
```

Execution:

./bin/prm_maker.exe

➔ "lstocdm2.prm" is created in "spacwkf/prm".

./bin/lstocdm2.exe

➔ All converted and separated files are stored in "./spacwkf/data/ls_files/Combined_Data".

All converted and separated files are automatically stored in "spacwkf/data/ls_files/Combined_Data".

Then, the subfolder "./spacwkf/data/cdm_files" is cleaned.

Finally, by the command "mv" at the last line all of the cdm files are moved from "./spacwkf/data/ls_files/Combined_Data" to "./spacwkf/data/cdm_files".

Example:

prm_maker.prm

prm_maker.prm (~/SPAC2016/SPAC2019/spacwkf/prm) - gedit

Open   Save

```
Testcase_   : site name (a9)
 4   3      : numbers of obs_ponts and channels
 30         : duration of each connected file in min.(integer)
17120516.45 : first file name (yymmddhh.mm)
  4         : number of output connected files
```

Plain Text   Tab Width: 8   Ln 5, Col 4   INS

lstocdm2.prm: automatically created by lstocdm2.sh

lstocdm2.prm (~/SPAC2016/SPAC2019/spacwkf/prm) - gedit

Open   Save

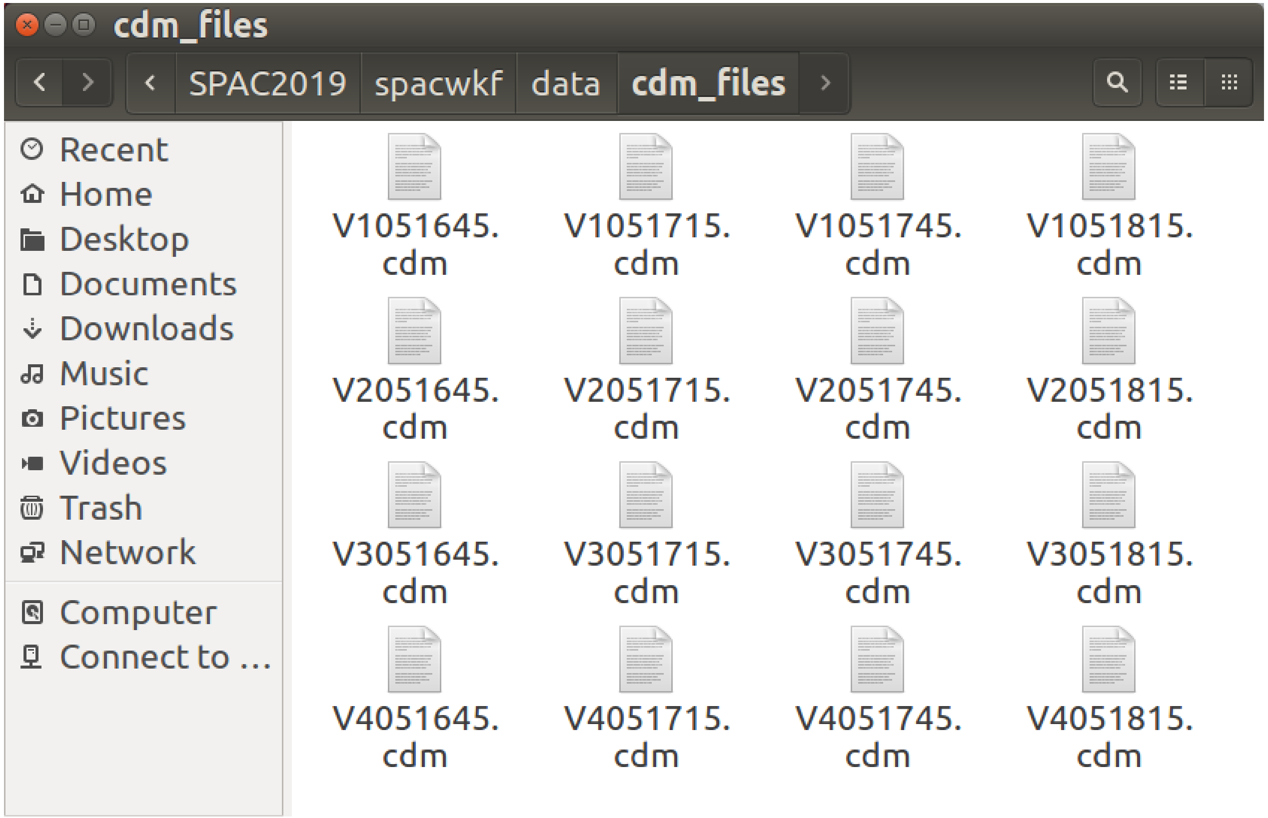
```
 4                                 : case numbers
                                   : This blank line is necessary
Testcase_no1                       : Data Folder Name(a12)
  794.   792.   798.               : #1 T44002 Sensitivity (V/M/s)
NS EW V                            : channel labels
0.15646E-06                        : voltage per Least Significant Bit (V/LSB)
V1     30                          : (a2) channel identifier of name; duration(min); of output file
 3 1 2 1                           : channel pivoting, channel selection (0=all, 1=1st channel only)
17120516.45    4                   : 1st File name(a11),number of output files
                                   : This blank line is necessary
Testcase_no2                       : Data Folder Name(a12)
  798.   798.   796.               : #2 T44003 Sensitivity (V/M/s)
NS EW V                            : channel labels
0.15646E-06                        : voltage per Least Significant Bit (V/LSB)
V2     30                          : (a2) channel identifier of name; duration(min); of output file
 3 1 2 1                           : channel pivoting, channel selection (0=all, 1=1st channel only)
17120516.45    4                   : 1st File name(a11),number of output files
                                   : This blank line is necessary
Testcase_no3                       : Data Folder Name(a12)
  796.   794.   798.               : #3 T44004 Sensitivity (V/M/s)
NS EW V                            : channel labels
0.15646E-06                        : voltage per Least Significant Bit (V/LSB)
```

Plain Text   Tab Width: 8   Ln 1, Col 1   INS

```
yokoi@ubuntu: ~/SPAC2016/SPAC2019
File Edit View Search Terminal Help
yokoi@ubuntu:~/SPAC2016/SPAC2019$ sh lstocdm2.sh
 Parameter file: ./spacwkf/prm/lstocdm2.prm

Name of the site: Testcase
No. of station: no1
 Sensitivity(M/s):    794.000000       792.000000       798.000000
 Components: NSEWV
 voltage per Least Significant Bit   1.56460004E-07 (V/LSB)
   1.97052905E-05   1.97550507E-05   1.96065157E-05
Output 1 channel  V
Start from: 17120516.45 ,    30 file
 Output file: ./spacwkf/data/ls_files/Combined_Data/V1051645.cdm

 Input file: ./spacwkf/data/ls_files/no1/17120516.45

 Output file: ./spacwkf/data/ls_files/Combined_Data/V1051715.cdm

 Input file: ./spacwkf/data/ls_files/no1/17120517.15

 Output file: ./spacwkf/data/ls_files/Combined_Data/V1051745.cdm

 Input file: ./spacwkf/data/ls_files/no1/17120517.45

 Output file: ./spacwkf/data/ls_files/Combined_Data/V1051815.cdm

 Input file: ./spacwkf/data/ls_files/no4/17120517.15

 Output file: ./spacwkf/data/ls_files/Combined_Data/V4051745.cdm

 Input file: ./spacwkf/data/ls_files/no4/17120517.45

 Output file: ./spacwkf/data/ls_files/Combined_Data/V4051815.cdm

 Input file: ./spacwkf/data/ls_files/no4/17120518.15

yokoi@ubuntu:~/SPAC2016/SPAC2019$
```

```
yokoi@ubuntu: ~/CCA2017/CCA2017/lstocdm2
All zero for 0.5byte data 15 12 29 32 ch- 0      0.0
All zero for 0.5byte data 15 12 29 33 ch- 0      0.0
All zero for 0.5byte data 15 12 29 34 ch- 0      0.0
All zero for 0.5byte data 15 12 29 35 ch- 0      0.0
All zero for 0.5byte data 15 12 29 36 ch- 0      0.0
All zero for 0.5byte data 15 12 29 37 ch- 0      0.0
All zero for 0.5byte data 15 12 29 38 ch- 0      0.0
All zero for 0.5byte data 15 12 29 39 ch- 0      0.0
All zero for 0.5byte data 15 12 29 40 ch- 0      0.0
All zero for 0.5byte data 15 12 29 41 ch- 0      0.0
All zero for 0.5byte data 15 12 29 42 ch- 0      0.0
All zero for 0.5byte data 15 12 29 43 ch- 0      0.0
All zero for 0.5byte data 15 12 29 44 ch- 0      0.0
All zero for 0.5byte data 15 12 29 45 ch- 0      0.0
All zero for 0.5byte data 15 12 29 46 ch- 0      0.0
All zero for 0.5byte data 15 12 29 47 ch- 0      0.0
All zero for 0.5byte data 15 12 29 48 ch- 0      0.0
All zero for 0.5byte data 15 12 29 49 ch- 0      0.0
All zero for 0.5byte data 15 12 29 50 ch- 0      0.0
```

This error message means the clipping of data. Check the time and eliminate the corresponding part.



File name (E1151145.cdm) includes the following information:

1st letter: component

2nd letter: numbering of seismograph (=numbering of station)

3rd & 4th: Date in (i2)

5th & 6th: hour in (i2)

7th & 8th; minutes in (i2)

Example of a converted file

V1051645.cdm (~/SPAC2016/SPAC2019/spacwkf/data/cdm_files) - gedit

Open ▾ Save

```
File=17120516.45_17120517.14
                                    UNIT=mkine
                                                V
       0 00:00:00.000   0.1282057E+01
       1 00:00:00.010   0.8995534E+00
       2 00:00:00.020   0.1339935E+01
       3 00:00:00.030   0.2278970E+01
       4 00:00:00.040   0.3079465E+01
       5 00:00:00.050   0.2442665E+01
       6 00:00:00.060   0.8888678E+00
       7 00:00:00.070  -0.3812031E+00
       8 00:00:00.080  -0.8611901E+00
```

3 lines for header

Data lines:

Numbering, (A8) time, (A13) data (e16.7)

The created single channel file by this format conversion program will be read in the next step using “multipx6.exe” as follows.

<u>“multipx6.for</u>”

```
      character cline(3)*80,cdum*13,cdummy*8  ! Declare three
                                               ! character strings
…………………………………………………………..
      do i=1,3
        read(1,'(a80)')cline(i)   ! Read 3 lines header
      enddo
      do i=1,nst0-1
       read(1,*)cdummy            ! Skip first nst0 sec data
      enddo
      do i=1,ndur0
c read input data
        read(1,*,end=10)cdummy,cdum,xdum
         x(i,j)=xdum/scale
       enddo
   10  ndur0=i-1                   ! Adjust number of samples
…………………………………………………………..
```


Character strings cline(3) are not used further. Neither cdummy nor cdum.

Preparation for Multiplexing

- In the next step, *multipx6.sh* is used with the parameter file "*multipx6.prm*".
- For automatic editing of multipx6.prm, *multi_pre.sh* is prepared.
- Edit the following "*multi_pre.prm*" in "spacwkf/prm" and run *"sh multi_pre.sh*".

multi_pre.prm (~/SPAC2016/SPAC2019/spacwkf/prm) - gedit

Open ▾ Save

```
1 TESTCASE              Dec/05/17:00-15H,2017        : comment(A50)
2  30          : duration of each connected file in min.(integer)
3 V1051645     : first file name (yymmddhh.mm)
4 cdm          : extention (A3)
5 1 2 3 4      : station pivot
6   4          : number of output connected files(integer)
```

- 1st line: Comment but later used as the title of all graphs showing the results of analysis.
- 2nd line: Duration of connected files same as the 3rd line of "prm_maker.prm".
- 3rd line: The eariest file name for the 1st position (A8)
- 4th line: Extension of the name of files in the folder "cdm_files", i.e., cdm.
- 5th line: Station pivoting list.
- 6th line: Number of connected files same as the 5th line of "prm_maker.prm".

Execution

```
yokoi@ubuntu:~/SPAC2016/SPAC2019$ sh multi_pre.sh
  Working Folder=./spacwkf/prm/
  Default Settings:
    dt       = 0.0
    np       =   4
    tst      = 0.0
    file_out=CC**.dat

V1051645 V2051645 V3051645 V4051645
V1051715 V2051715 V3051715 V4051715
V1051745 V2051745 V3051745 V4051745
V1051815 V2051815 V3051815 V4051815
```

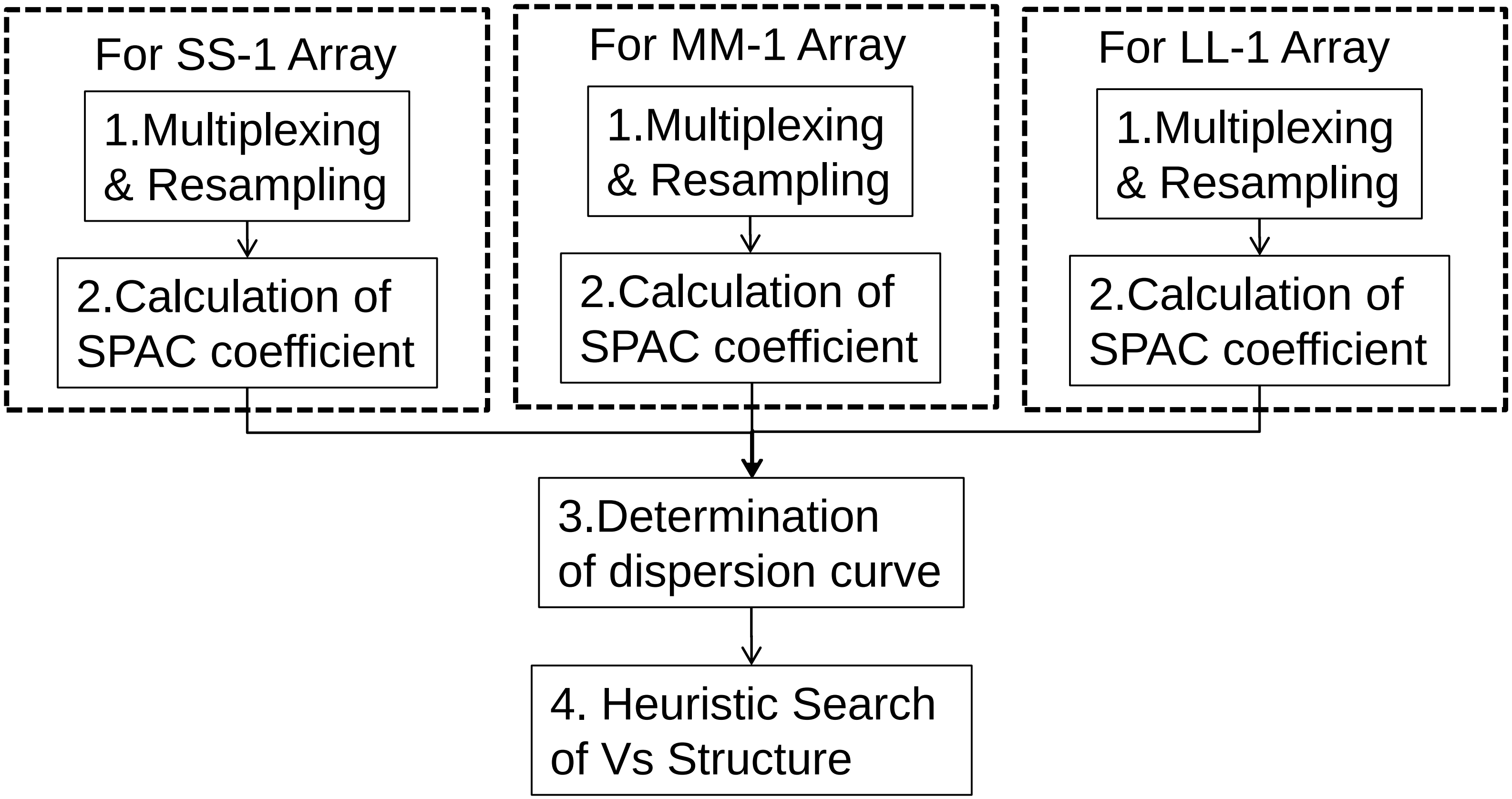
Automatically created multipx6.prm which will be used in the next step.

```
multipx6.prm (~/SPAC2016/SPAC2019/spacwkf/prm) - gedit
Open ▾                                                          Save

    1                             : Number of cases
                                  : This blank is necessary
  4 0.010                         : Number of obs points,dt
          0.0       1800.0        : tst,tdur(sec)
1.e0  mkine                       : scale(input data is divided by this scale)
 0 3 0.1 1.0 1.5                  : nfilter(=1:apply), ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
  4          .cdm                 : nattach, cattach
 2   CC                           : n_out(A12),cout ("**.dat" is attached)
TESTCASE             Dec/05/17:00-15H,2017         : comment(A50)
    4            8                : number of measurement in the same array configuration,n_character
V1051645 V2051645 V3051645 V4051645
V1051715 V2051715 V3051715 V4051715
V1051745 V2051745 V3051745 V4051745
V1051815 V2051815 V3051815 V4051815

                          Plain Text ▾  Tab Width: 8 ▾   Ln 11, Col 36  ▾  INS
```

Edit it in an appropriate way if necessary.

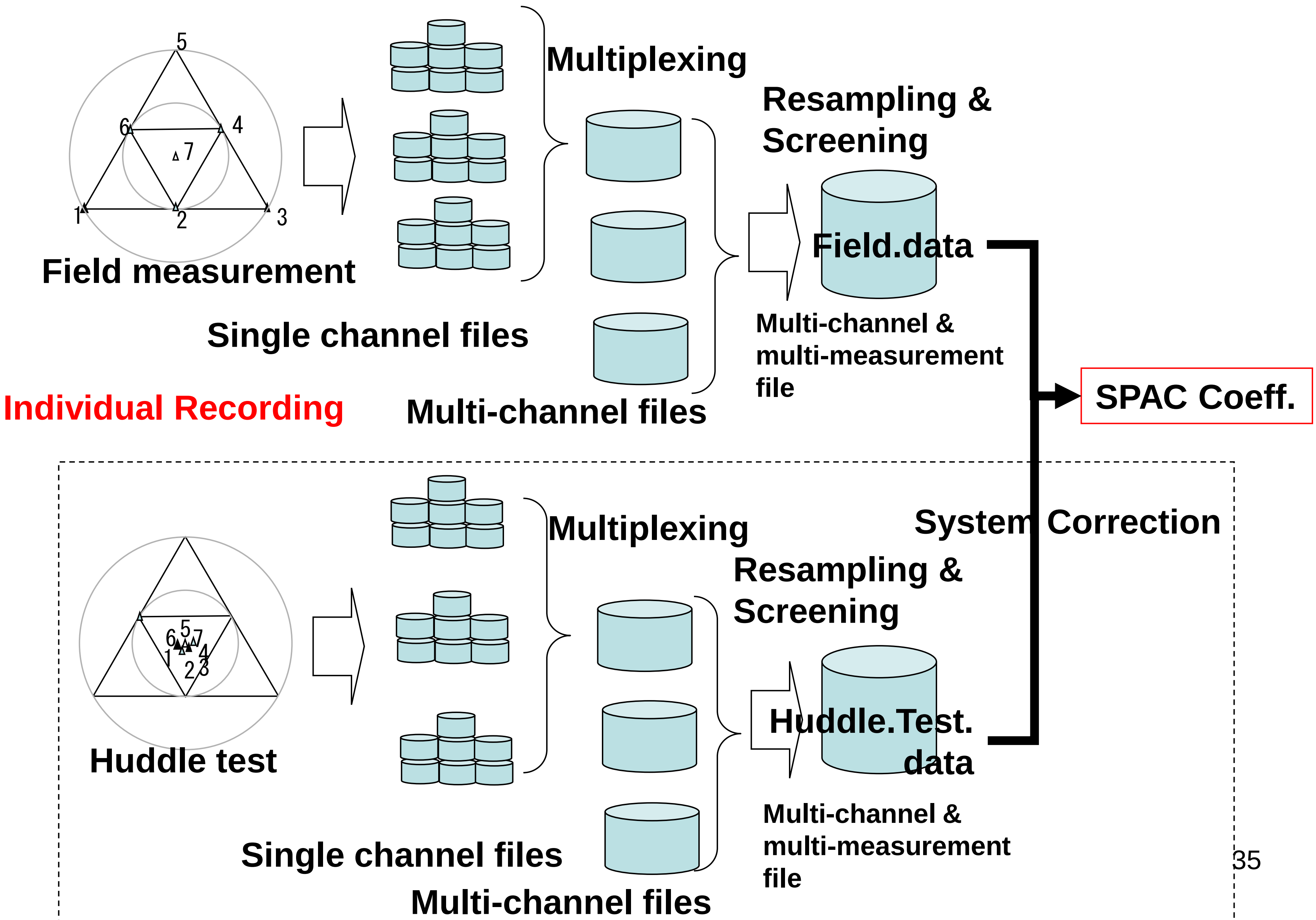
Note: Example

An example that consists of two arrays of different sizes is shown below.
+ SS-1 Array: 7 points equilateral triangle array with the center
+ MM-1 Array: 4 points equilateral triangle array with the center
+ LL-1 Array: 4 points equilateral triangle array with the center



1. (Multiplexing &) Resampling

Flow of the data processing for the conventional SPAC



1. Multiplexing & Resampling

1.1. Multiplexing (Optional)

Single channel files are combined into a multi-channel file
(This step is not necessary for the multi-channel recording cases)

multipx6.sh + ./spacwkf/prm/multipx6.prm

First step is to edit the parameter file multipx6.prm.

However, it is recommendable for users to make own conversion program from her/his original format files directly to multiplexed files of the format explained below.

Terminology

Multiplexing:

To sort the data individually stored in single channel files into a multi-channel file of the time-sequential format.

Huddle test:

Common input motion recording to determine the difference of the system characteristics among the recording system and/or channels.
The seismometers used in field measurement are put close each other like a huddle and simultaneous recording is conducted.

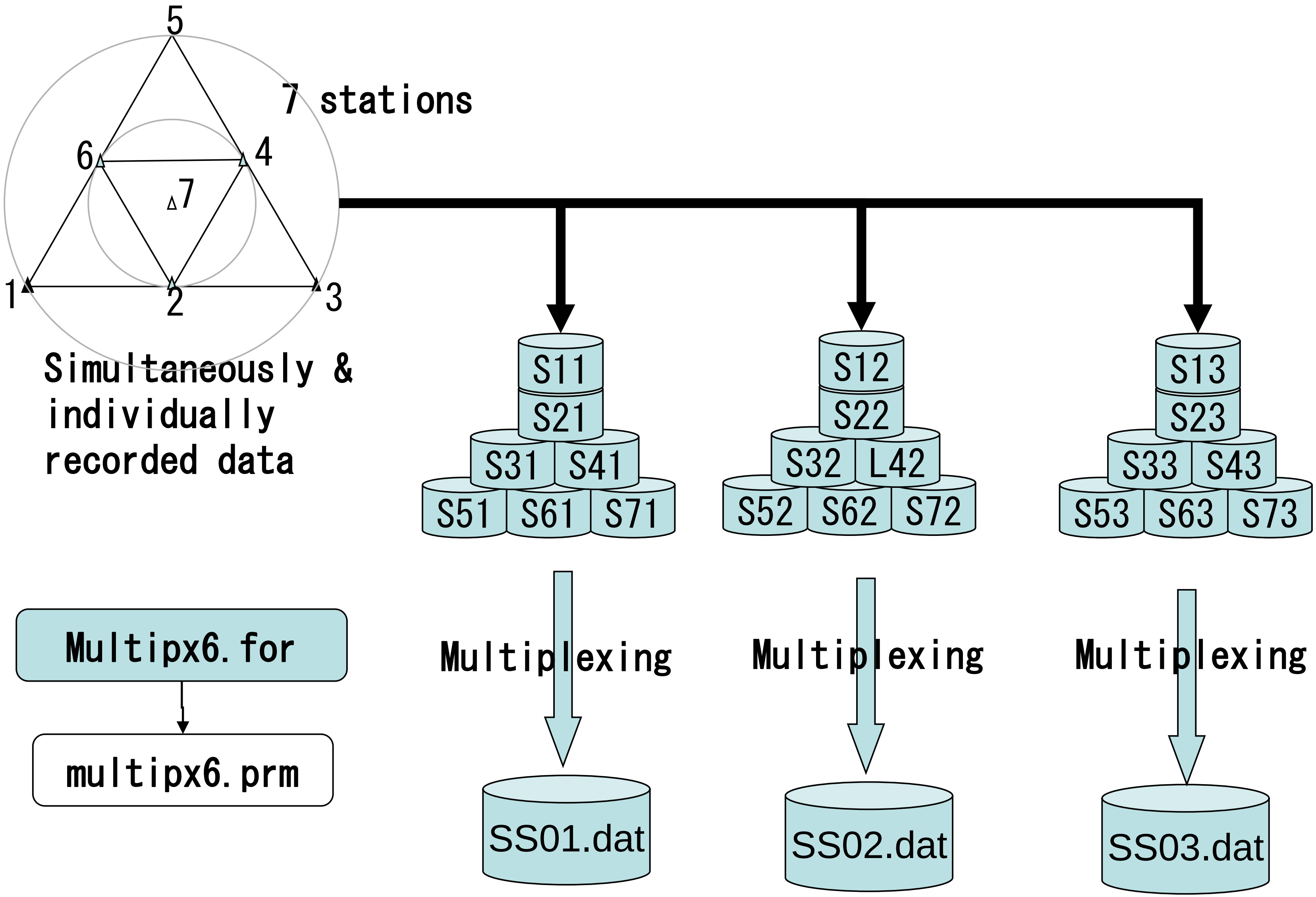
Re-sampling:

It can be done to thin the data out in order to reduce the size of data files and the load to PC for processing. This can cause the aliazing effect. Then, it Is necessary to apply the digital anti-alias filter that has high cut characteristics before thinning out.

System correction:

The difference of the characteristics among the recording system can be corrected using the data obtained by huddle tests.

Multiplexing the single channel files



Multi-channel files

Warning:

The program multipx6.sh does not have the functionality to adjust the time difference among the single channel files. The input files must have the same timing.

Confirm the following:
+ All input files have the same sampling interval.
+ All input files have their first datum that are sampled at the same time.

multipx6.prm

1st case:

```
 3                          :Number of cases
                            :This blank is necessary
 4 0.01                     :Number of Channels,dt
0.0   163.83                :tst,tdur
1.e11  mkine                :scale(input data is divided by this scale),unit(a5)
0 3 0.005 1.0 1.5           :nfilter(=1:apply), ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
 4   .cdm                   :nattach, cattach
2 MM                        :n_out(A12),cout    ("**.dat" is attached)
MM-1, Iwaki City Hall, Dec.22, 2012                    :comment(A50)
21 8                        :number of measurement in the same array configuration,n_character
sxm21001 sxm21002 sxm21003 sxm21004
...
sxm23601 sxm23602 sxm23603 sxm23604
```

2nd case:

```
                            :This blank is necessary
 7 0.002                    :Number of Channels,dt
0.0    32.766               :tst,tdur
3.16e2  mkine                 :scale(input data is divided by this scale)
0 3 0.005 1.0 1.5           :nfilter(=1:apply), ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
 4   .cdm                   :nattach, cattach
2 SS                        :n_out(A12),cout    ("**.dat" is attached)
SS-1, Iwaki City Hall, Dec.22, 2012                    :comment(A50)
30 8                        :number of measurement in the same array configuration,n_character
sxs10101 sxs10102 sxs10103 sxs10104 sxs10105 sxs10106 sxs10107
…
sxs13001 sxs13002 sxs13003 sxs13004 sxs13005 sxs13006 sxs13007
```

3rd case:

```
                            :This blank is necessary
 4 0.01                     :Number of Channels,dt
0.0   163.83                :tst,tdur
3.16e10  mkine                :scale(input data is divided by this scale)
0 3 0.005 1.0 1.5           :nfilter(=1:apply), ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
 4   .cdm                   :nattach, cattach
2 LL                        :n_out(A12),cout    ("**.dat" is attached)
LL-1, Iwaki City Hall, Dec.22, 2012                    :comment(A50)
28 8                        :number of measurement in the same array configuration,n_character
sxl21001 sxl21003 sxl21005 sxl21007
…
sxl24601 sxl24603 sxl24605 sxl24607
```

multipx6.prm

```
 3                    :Number of cases
                      :This blank is necessary
 4 0.01               :Number of Channels,dt
0.0   163.83          :tst,tdur
1.e11  mkine          :scale(input data is divided by this scale),unit(a5)
0 3 0.1 1.0 1.5 :nfilter(=1:apply),ncharacter(=2:lowpass,=3:bandpass),fl,fh,fs
 4   .cdm             :nattach, cattach ➔ Input single channel file name
2 MM                  :n_out(A12),cout ("**.dat" is attached)
MM-1, Iwaki City Hall, Dec.22, 2012                   :comment(A50)
21 8        :number of measurement in the same array configuration,n_character
sxm21001 sxm21002 sxm21003 sxm21004 ← 1st
...
sxm23601 sxm23602 sxm23603 sxm23604 ← 21th measurement
   ↑        ↑        ↑        ↑
  1ch      2ch      3ch      4ch
```

Input file names : sxm2??0?.cdm

consist of the character string ‘sxm2??0?’ of 8 characters plus another character string ‘.cdm’ of 4 characters. These character strings and their number of characters are indicated in the 7th line for the latter and the 11th line and below for the former. Program ‘multipx6.exe’ automatically combines them and read the data from the files.

multipx6.prm

Output file name: SS01.dat for the 1st measurement. ’01’ shows the numbering of measurement.

…

SS30.dat for the 30th measurement. ’30’ shows the numbering of measurement.

These output file names consist of the character string ‘MM’ of 2 characters as indicated in the 8th line. The following two integers show the numbering of measurement. ‘.dat’ is attached to all automatically.

The data from tst to tst+tdur are processed in every files.

Values read from the input files are divided by the scale factor given in the 5th line. This value must be selected to make the unit of data in the output file is ‘mkine’, i.e., 1.0E-5 M/sec for ground velocity. For ground acceleration ‘gal’, i.e., 1.0E-2 M/sec^2 should be used. Otherwise the amplitudes of the data will be erroneously shown in the output figures.

sxm21001.cdm:

Example: format of input file (dt=0.01 sec) of single channel data
in ./spacwkf/data/cdm_files

```
File=01001.cdm      ch. 1
                 unit
                  V
    0 00:00:00.000  -0.1396391E+11
    1 00:00:00.010  -0.1513392E+11
    2 00:00:00.020  -0.1472600E+11
    3 00:00:00.030  -0.1314799E+11
    4 00:00:00.040  -0.7706377E+10
    5 00:00:00.050  -0.3947473E+10


16383 00:02:43.830  -0.7844215E+10
```

3 lines for header

Data lines:

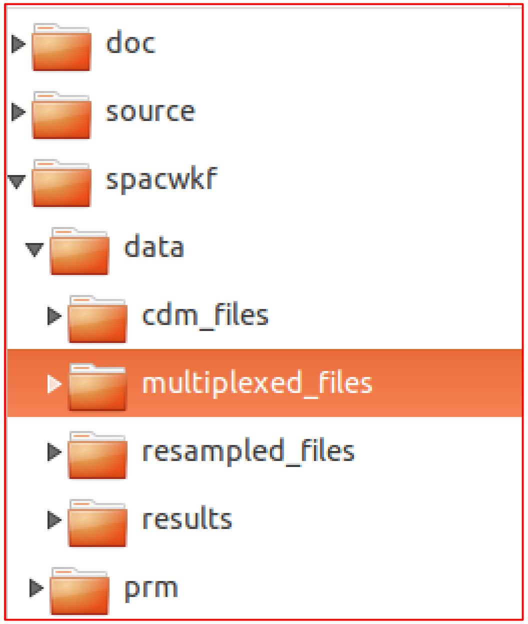
Numbering, (A8)   time, (A13)   data (e16.7)

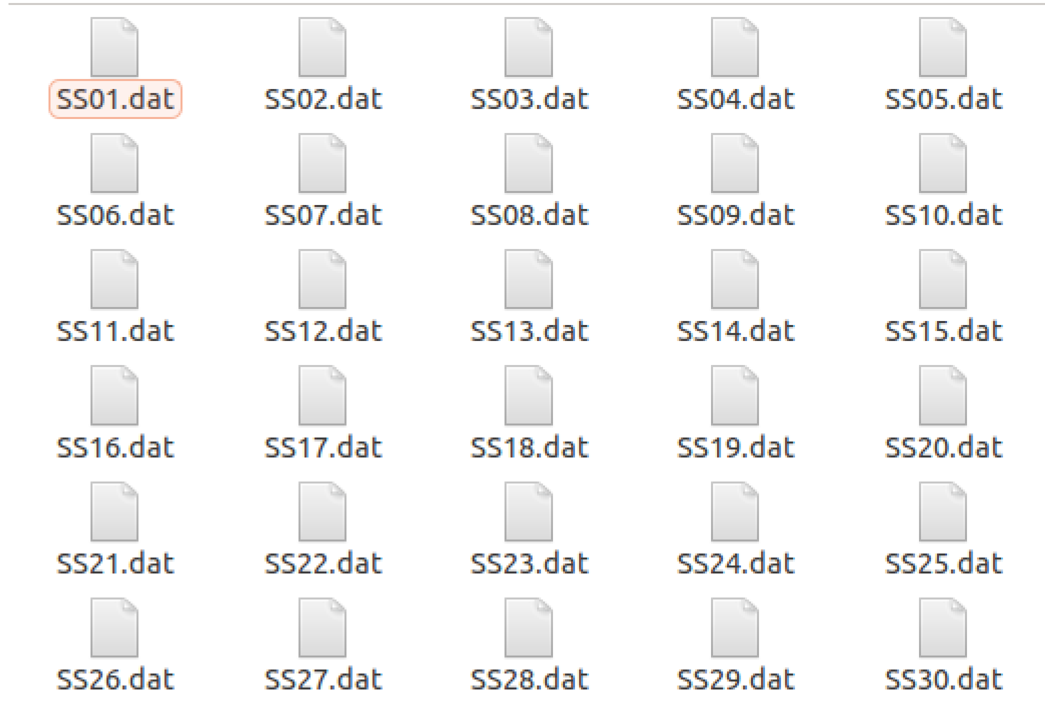
Example of execution (from the folder SPAC2019):

~$sh multipx6.sh

```
 Working Folder=./spacwkf/prm/
  Nch=           4  dt=   9.99999978E-03
  Scale Factor=   9.99999980E+10
  Does this scale convert the unit of data mkine ?
  Please change the value if not.
           0           3   4.99999989E-03   1.00000000       1.50000000
 MM-1, Iwaki City Hall, Dec.22, 2012              :
           1 -th measurement:           4
sxm21001 sxm21002 sxm21003 sxm21004
       16383



 ./spacwkf/data/multiplexed_files/MM01.dat
           2 -th measurement:           4
…
 Normal End
```

Example of output file (stored in the subfolder ./spacwkf/data/multiplexed_files






```
  7  0.0020      0.3160E+03   16382  mkine
SS-1, Iwaki City Hall, Dec.22, 2012                   :
    0.000000 -0.0000000E+00 -0.0000000E+00 -0.0000000E+00  0.0000000E+00 -0.0000000E+00  0.0000000E+00 -0.0000000E+00
    0.002000 -0.8961562E-04  0.5361709E-05 -0.2044998E-04  0.1081865E-03 -0.1978091E-05  0.2530022E-04 -0.4171700E-04
    0.004000 -0.1640327E-03  0.8138977E-04 -0.1974745E-04  0.1997052E-03  0.2068763E-04  0.4874512E-04 -0.9626019E-04
…
```

1st line: channel number, dt, scale factor, number of samples, unit
2nd line & below: time, 1ch,2ch,3ch,…, 7ch.

Warning!

- The declared array size for input data in multipx6.for is 1,200,000. This gives the constraint: tdur/dt < or =1,200,000
- Similarly nch< or =15

For much longer data file:

“multipx6.for” can handle 1200,000 samples of 15 channels at once. If you have data of longer recording time, it is recommendable to separate them beforehand.
An alternative may be the following way of using “tst” and “tdur” in ”multipx5.prm” can let you utilize the data fully.

For the 1st operation:
```
   0.0    3600.0                :tst(start time)),tdur(duration) in sec.
```

For the 2nd operation:
```
3600.01   3600.0                :tst(start time)),tdur(duration) in sec.
```

For the 3rd operation:
```
7200.01   3600.0                :tst(start time)),tdur(duration) in sec.
```
…

Namely, “multipx6.for”skips int(tst/dt) data and then starts reading the next data.

Example Data files: These are already multiplexed.

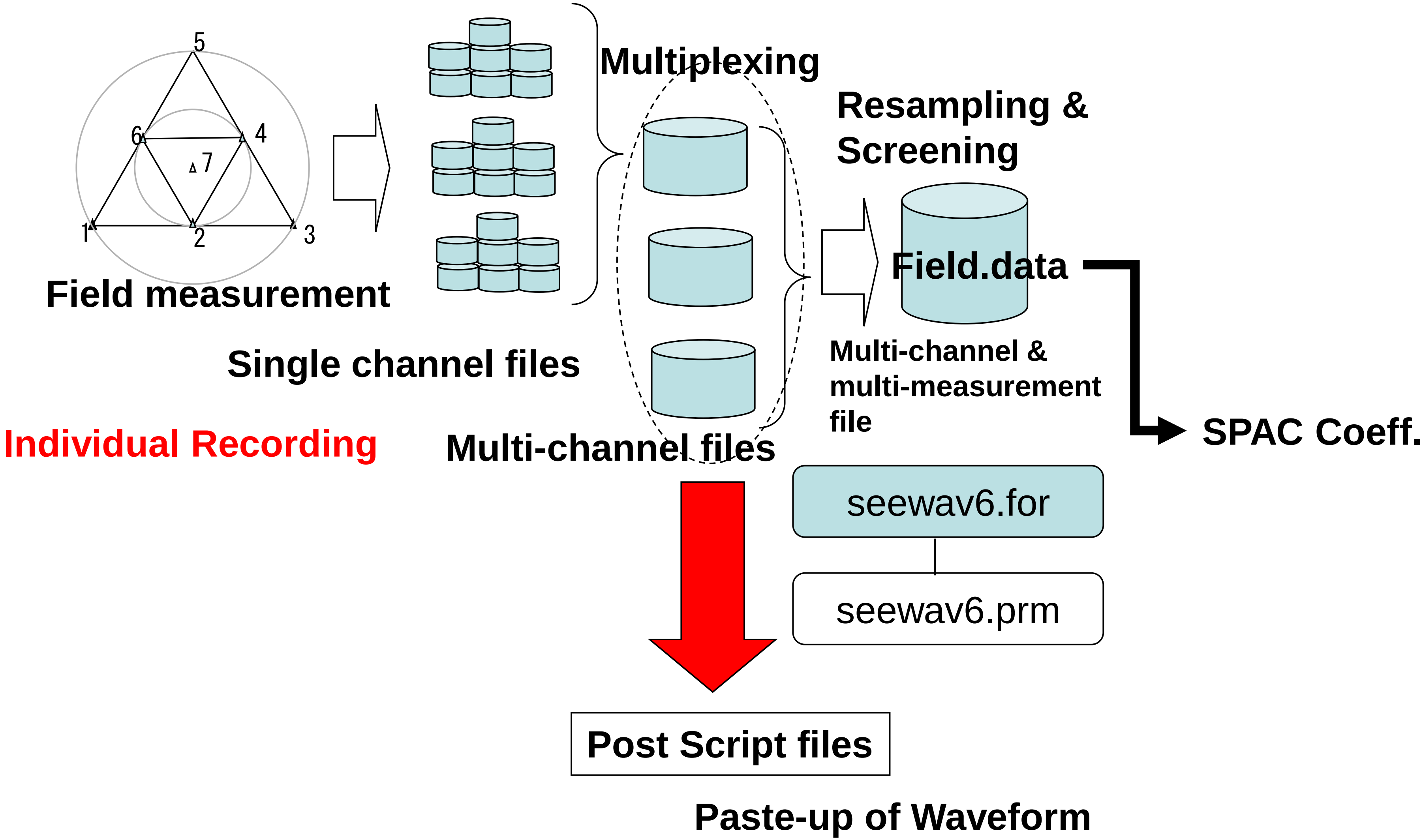
multi_SS.tar.gz
multi_MM.tar.gz
multi_LL.tar.gz

Copy these three compressed files into the subfolder "./spacwkf/data/multiplexed_files" for exercise with these example Data files.

1.2. Plot Waveform

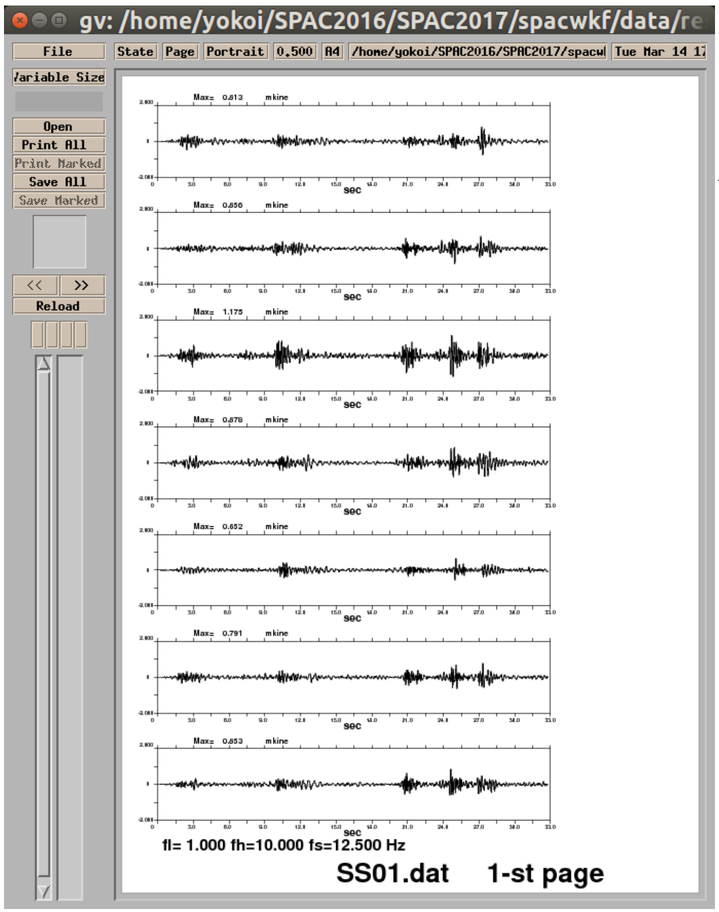
seewav6.sh+ ./spacwkf/prm/seewav6.prm

Flow of the data processing for the conventional SPAC



Multi-channel file: SS01.dat

```
 7  0.0020       0.3160E+03    16382  mkine
SS-1, Iwaki City Hall, Dec.22, 2012                      :
     0.000000 -0.0000000E+00 -0.0000000E+00 -0.0000000E+00  0.0000000E+00 -0.0000000E+00  0.0000000E+00 -0.0000000E+00
     0.002000 -0.8961562E-04  0.5361709E-05 -0.2044998E-04  0.1081865E-03 -0.1978091E-05  0.2530022E-04 -0.4171700E-04
     0.004000 -0.1640327E-03  0.8138977E-04 -0.1974745E-04  0.1997052E-03  0.2068763E-04  0.4874512E-04 -0.9626019E-04
...
```



seewav6.for

seewav6.prm

SS01.ps

Figures in Multi-page Post Script file. Post Script file can be opened, for example, by "gv &", where "gv and "&" stand for "ghost view" and background operation.

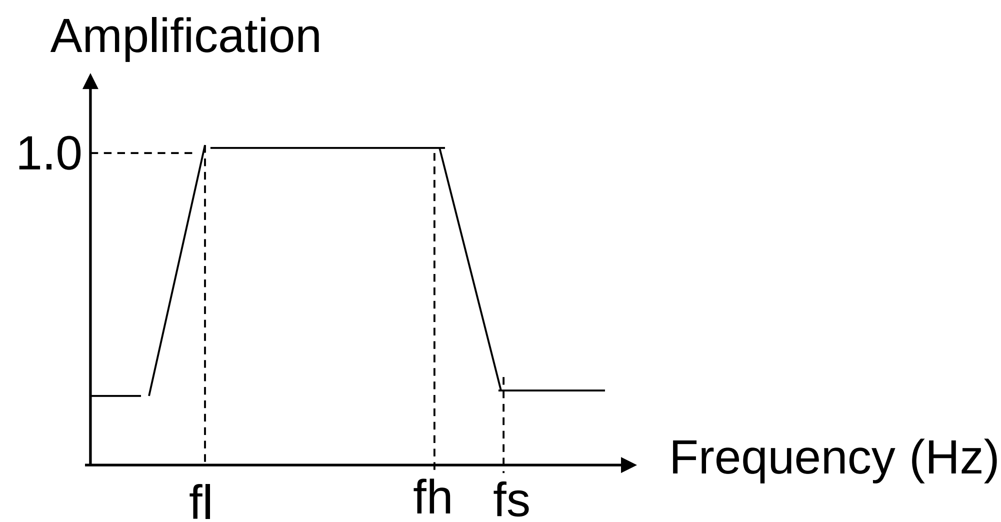
If "ghost view" is not installed yet:
sudo apt-get install gv

seewav6.prm

dtl denotes the time duration that corresponds to 1cm along the time axis.
In one page, 28*dtl/dt time step can be plotted. If the file has more, new pages are automatically added as much as necessary and multi-page PS file is created.

```
 7                        :nch
1  1.0 10.0 12.5 3        :nfilter,fl,fh,fs,nchara(=2:lowpass, =3:bandpass)
  1.5                     :dtl(sec/cm),25,50==>10,20 min/page
30 8                      :n_mea,n_character
SS01.dat
SS02.dat
...
SS30.dat
```

Band Pass Filter



nfilter=0: no effect
nfilter=1: bandpass filter is applied

This BPF does not affect to the data files.

If the records have significant DC-offset (base line shift due to DC component)

seewav6.prm

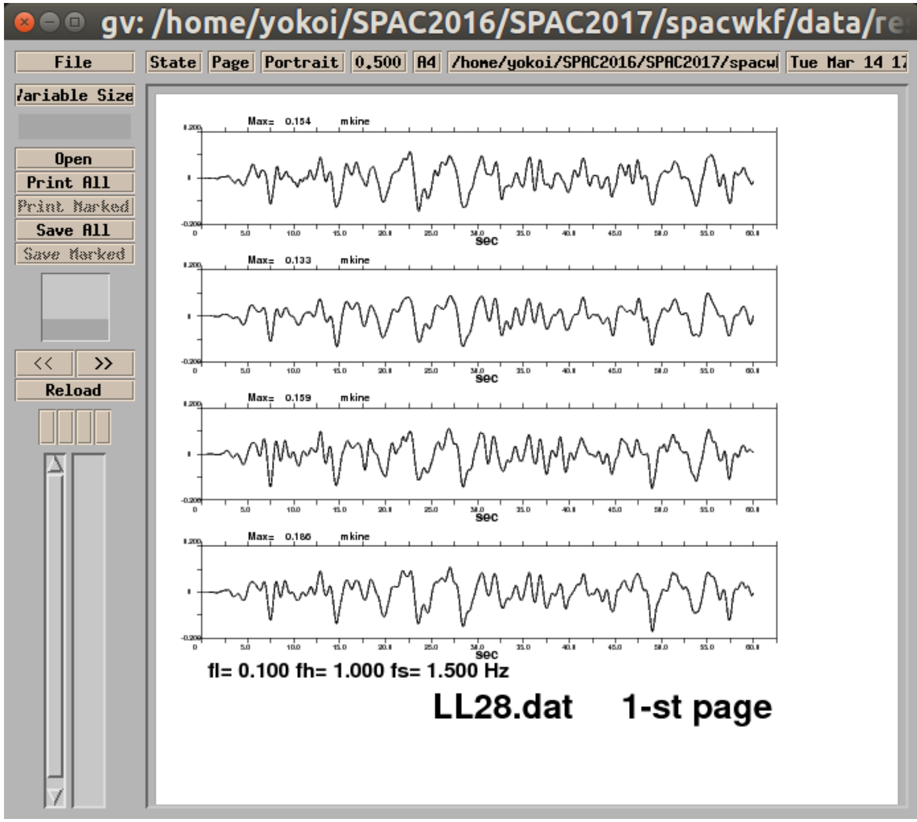
```
  7                         :nch
1   0.1  50.   60.0         :nfilter,,fl,fh,fs
  1.0                       :dtl(sec/cm)
30 8                        :n_mea,n_character
SS01.dat
...
SS30.dat
```

Set nbandpass = 1 and fs>Nyquist frequency

Waveform plots with DC-offset correction, but without applying bandpass filter, are given.

Example of execution:

```
$ sh seewav6.sh
 ./spacwkf/prm/seewav6.prm
           1  0.100000001       1.00000000       1.50000000              3
          28           8
 ./spacwkf/data/multiplexed_files/LL01.dat
  DC-offset removed & tapered;
  0.154889539      0.200000003
./spacwkf/data/results/fig_wave/LL01.ps
LL01.dat     1-st page
…
 ./spacwkf/data/multiplexed_files/LL28.dat
  DC-offset removed & tapered;
  0.186059728      0.200000003
./spacwkf/data/results/fig_wave/LL28.ps
LL28.dat     1-st page
~/SPAC2019 $
```



Output files:

PostScript files are stored in the folder:

./spacwkf/data/results/fig_wave

with extension "ps". Use "gv &" to draw it.

1.3. Re-sampling & Screening

Re-sampling & Screening

resample5.sh + ./spacwkf/prm/resample5.prm

or

resample6.sh + ./spacwkf/prm/resample6.prm

Then, check the selected time blocks using

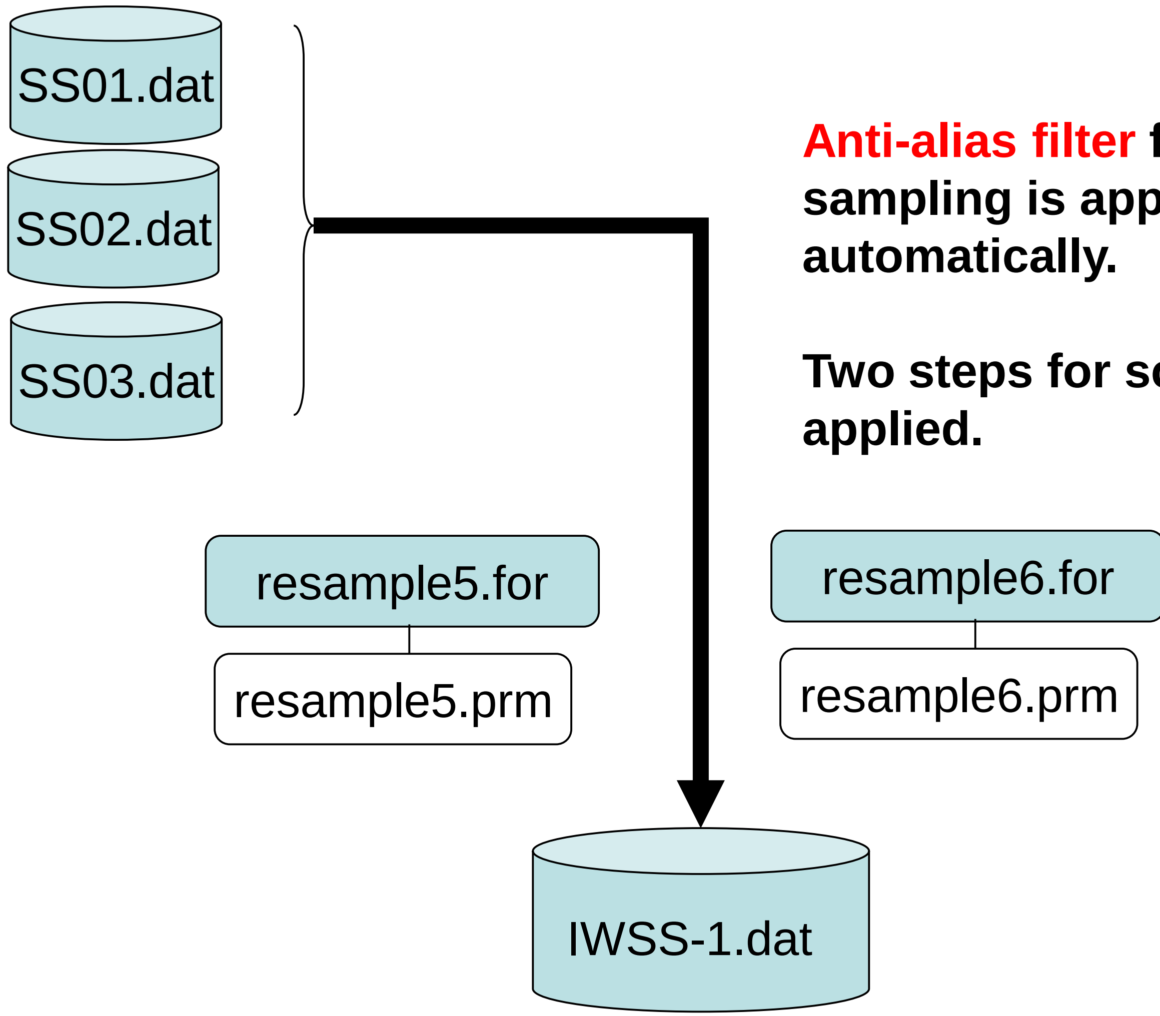
seeblk1.sh + seeblk1.prm

Note: Modify the file “./spacwkf/prm/graph_title.txt”, if you have not started the processing from “*sh seg2read.sh*”. This means that your original data files are not seg2 format.

The contents of the file “./spacwkf/prm/graph_title.txt” is used for the title of various graphs produced in further processing. It is recommendable to give an appropriate title to the figures to prevent potential confusion.

Re-sampling & Screening

Multi-channel data files from the same array configuration



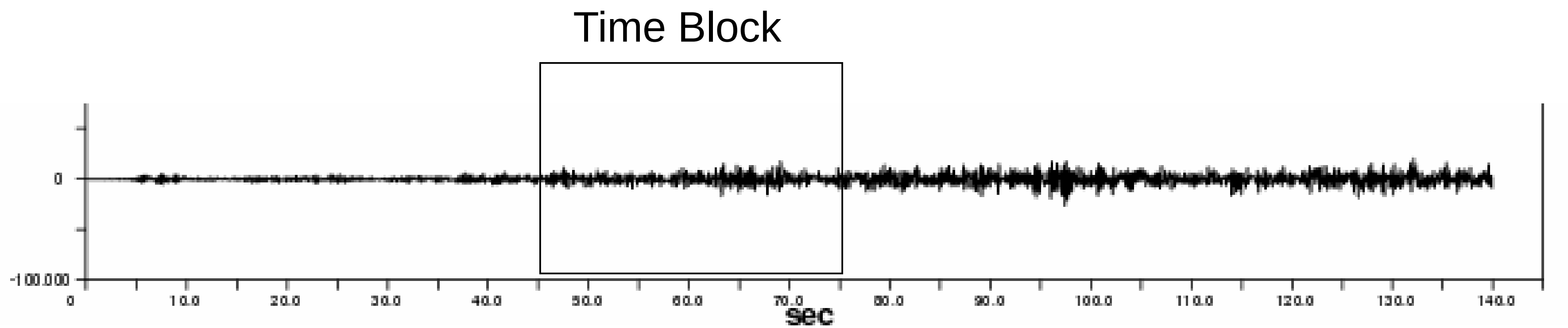
Anti-alias filter for re-sampling is applied automatically.

Two steps for screening are applied.

Resampled & screened multi-channel & multi-measurement data file

Screening: Step-1

Parameter: ajudge



If the maximum amplitude in a time block exceeds the product of the “ajudge” parameter to RMS amplitude of the same time block, this time block is not used in analysis.
This is a countermeasure against impulsive noise due to traffic, i. e., vehicles passing near by seismometers.
The bigger value of “ajudge” means looser screening. The smaller value means fewer available time blocks.

Screening: Step-2

Parameter: a_sgm

If the RMS amplitude in a time block deviates more than the “a_sgm” parameter multiplied to the standard deviation from the average, this time block is not used in analysis, where the average and the standard deviation are calculated over the all time blocks that survived in the screening Step-1.

This is a countermeasure against outliers.
The bigger value of “a_sgm” means looser screening. The smaller value means fewer available time blocks.

Warning:

Be sure to use the same resampling interval and the same block size, in case of the combination of arrays of various sizes. Otherwise, a heavy problem will take place in the further step of analysis, namely, the determination of dispersion curves.

The declared array size for input data in resample5.for and resample6.for is 750,000.
This gives the constraint: tdur/dt < or =750,000
Similarly nch< or =11
and number of data in one time block < or = 8192

It is recommendable to make a multiplexed data file for every 30 minutes or shorter duration, in order to avoid the problem due to the exceedance of input data quantity.

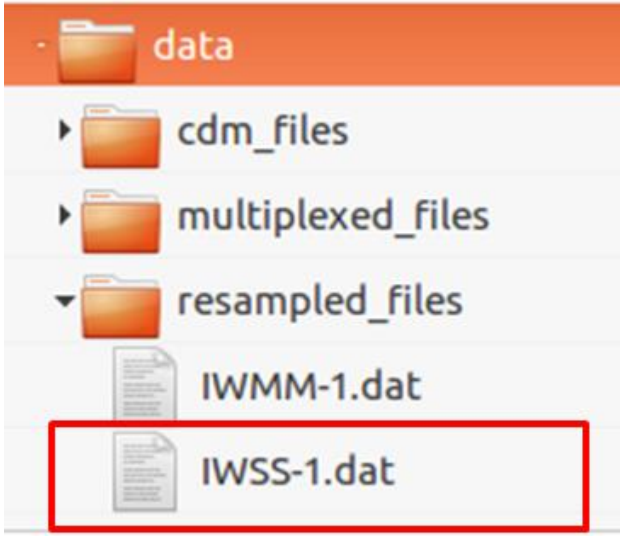
resample5.prm or resample6.prm

```
7 0.002 10          :nch,dt,number of Channels,dt,nskip
7.0 1.5             :ajudge,a_sgm
0.0  32.766         :tst,tdur
IWSS-1.dat          :output file name
1024                :number of data in one time block after resampling
30                  :number of measurement in the same array configuration
SS01.dat
SS02.dat
SS03.dat
…
SS30.dat
```

1st line: Channel number, sampling interval, skip data number
2nd line: ajudge, a_sgm: parameters for two step screening
3rd line: start time and duration for processing
4th line: Output file name
5th line: number of data in one time block after resampling.
resample5.for divides all the data into the time blocks that have this number of data allowing overlapping of 50 % between neighboring blocks.
6th line: number of measurement in the same array configuration.
Example1 is the single measurement case.
7th line: Input file name

Example of execution: Output file is stored in './spacwkf/data/resampled_files'

```
~/SPAC2019$ sh resample5.sh
 ./spacwkf/prm/resample5.prm
  Working Folder=./spacwkf/prm/
 Band-Pass:fl=  0.100000001     fh=   20.2499981     fs=   22.4999981
  Nch=           7
  Nskip=          10  f(Nyquist)=   24.9999981      fs=   22.4999981
  ajudge=   7.00000000      a_sgm=   1.50000000
   0.00000000       32.7659988
  First screening (peak/rms<   7.00000000     ):
           1 -th measurement:./spacwkf/data/multiplexed_files/SS01.dat
           2  blocks remained among            2  blocks
...
 30 -th measurement:./spacwkf/data/multiplexed_files/SS30.dat
           2  blocks remained among            2  blocks
  Data stored in the temporary file
 ./spacwkf/data/resampled_files/IWSS-1.dat
  Second screening ({rms-average(rms)}/sigma<   1.50000000     ):
          27  blocks remained among           39  blocks
~/SPAC2019$
```



Resampled & screened multi-channel & multi-measurement data file

1st line:file parameters

1st block: lines 1 … 1024 (first group); 2nd block: lines 1 … 1024 (second group)

```
       7      27       1024  (i8,f16.4, 7e15.7) mkine
       1          0.0000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
       2          0.0200 -0.6542959E-10  0.6730985E-10  0.8688813E-11 -0.9121746E-10  0.3278066E-10  0.3681669E-10 -0.2157925E-10
     …
    1024         20.4600  0.6798404E-01  0.8341893E-01  0.7360612E-01 -0.3876965E-01  0.1080229E+00 -0.9545929E-01 -0.1346539E+00
       1         10.2400 -0.1701506E+00  0.9676924E-01  0.3960467E-01 -0.3436972E+00  0.1931517E+00  0.1742720E-01  0.8249036E-01
     …
    1024         30.7000 -0.4059438E-01  0.2059561E-02  0.2788787E-01 -0.1173913E+00  0.2213649E-01 -0.1785946E-01 -0.9963673E-01
       1          0.0000  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
     …
```

The file includes mmblk=27 time blocks of nch=7 channel data.
Each time block is composed of nblk=1024 data.
Each line corresponds to a time step. The format used to store each line is cform3='(i8,f16.4, 7e15.7)'.
The unit for these data is "mkine".
These file parameters are stored in the 1st line.
As all of the data are delimited by space, this file can be read using free format.

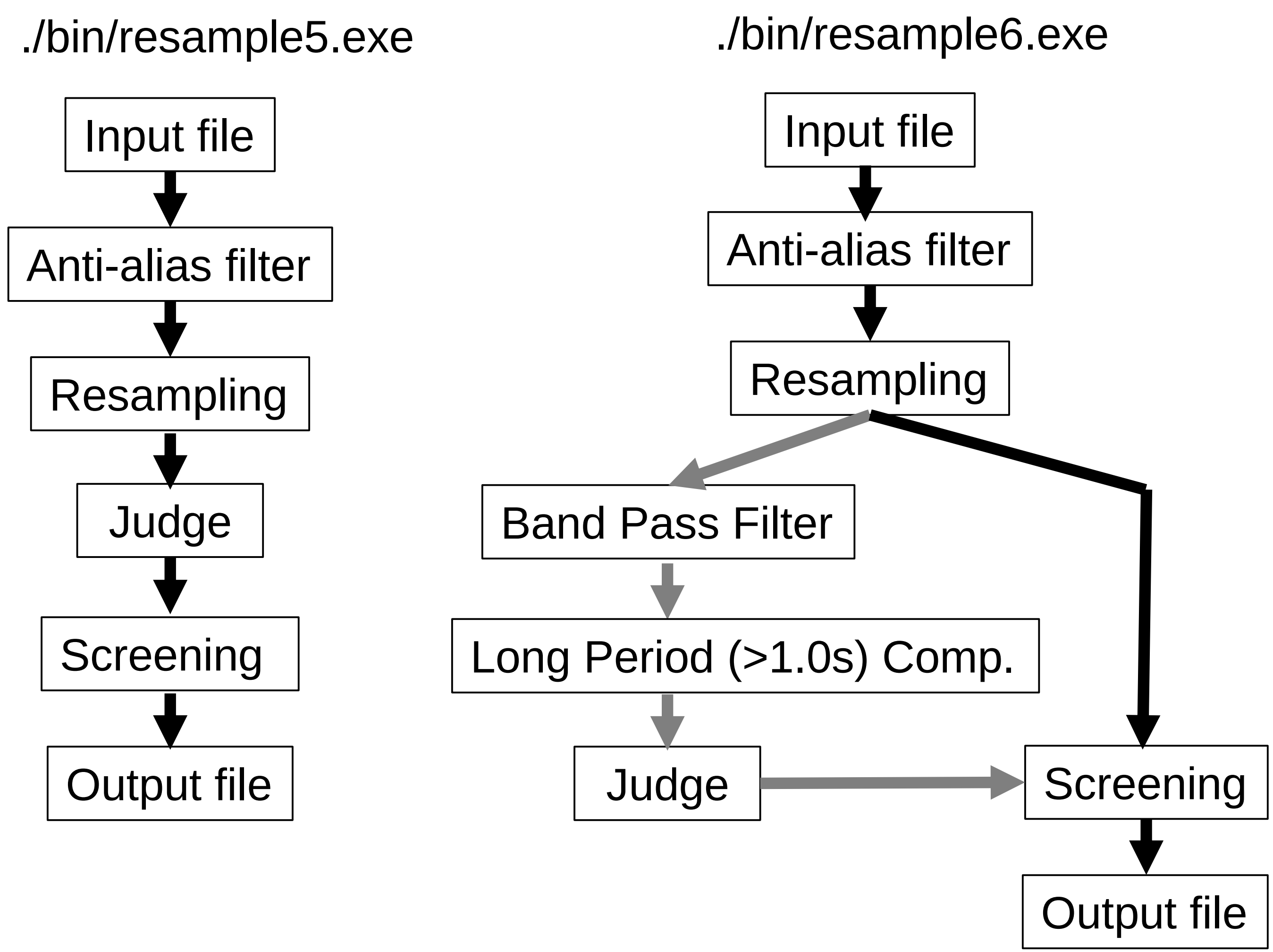
For MM-1 array: resample6.prm

```
4 0.01  2              :nch,dt,number of Channels,dt,nskip
4.0 2.0                :ajudge,a_sgm
0.0 163.83             :tst,tdur
IWMM-1.dat             :output file name
1024                   :number of data in one time block after resampling
21                     :number of measurement in the same array configuration
MM01.dat
…
MM21.dat
```

Execution:

```
~SPAC2019$ sh resample6.sh
 ./spacwkf/prm/resample6.prm
  Working Folder=./spacwkf/prm//
 Band-Pass:fl=  0.100000001     fh=   20.2500000     fs=   22.5000000
  Nch=           4
  Nskip=           2  f(Nyquist)=   25.0000000      fh=   20.2500000
  ajudge=   4.00000000      a_sgm=   2.00000000
   0.00000000       163.830002
  First screening (peak/rms<   4.00000000     ):
           1 -th measurement:./spacwkf/data/multiplexed_files/MM01.dat
 Band-Pass:fl1=   5.00000007E-02 fh1=   1.00000000     fs1=   1.50000000
          14  blocks remained among           14  blocks
…
 21 -th measurement:./spacwkf/data/multiplexed_files/MM21.dat
 Band-Pass:fl1=   5.00000007E-02 fh1=   1.00000000     fs1=   1.50000000
 14  blocks remained among           14  blocks
  Data stored in the temporary file
 ./spacwkf/data/resampled_files/IWMM-1.dat
  Second screening ({rms-average(rms)}/sigma<   2.00000000     ):
         280  blocks remained among          294  blocks
~SPAC2019$
```

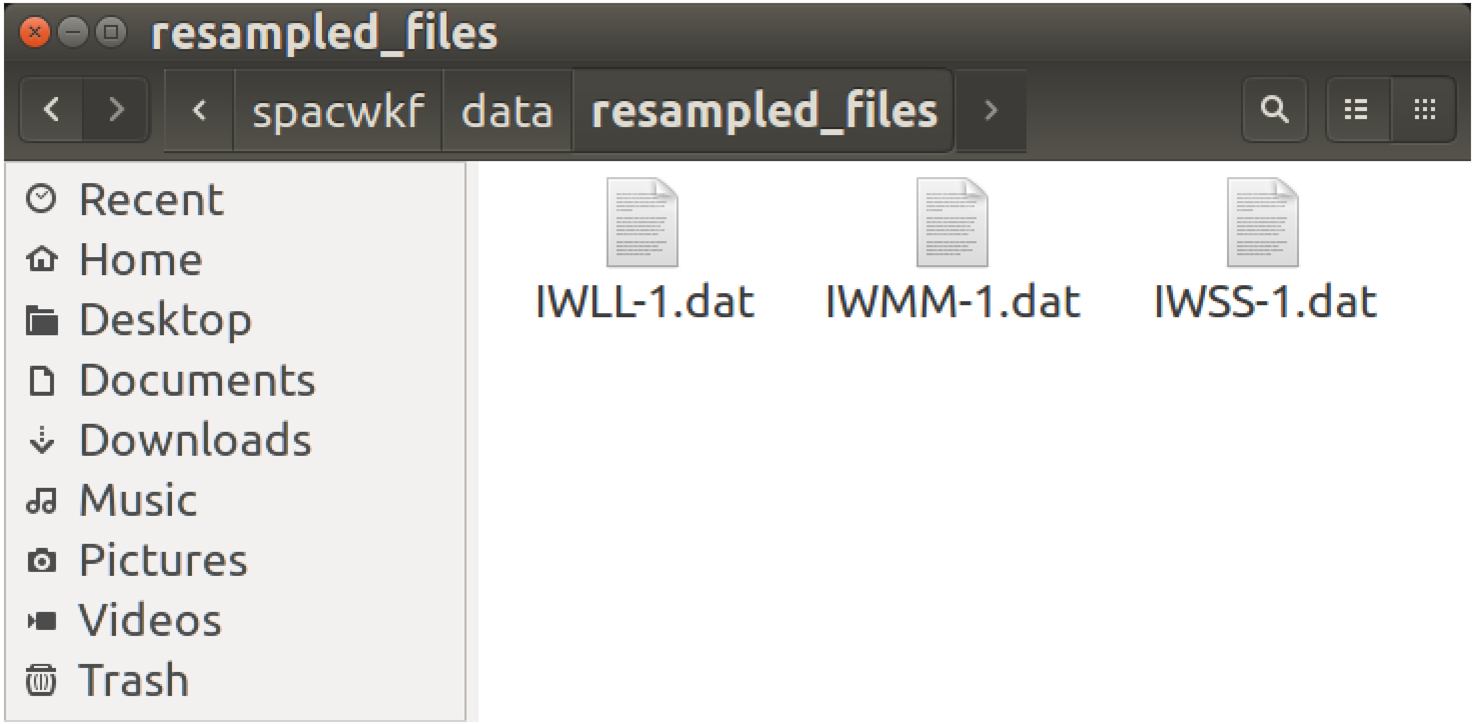
What's special of resample6.exe in comparison with resample5.exe



For short period observation

For long period observation

```
Output files:
IWSS-1.dat ← resample5.exe
IWMM-1.dat ← resample6.exe
IWLL-1.dat ← resample6.exe

Resample6.prm is edited after obtaining IWMM-1.dat
```



Check the selected time blocks

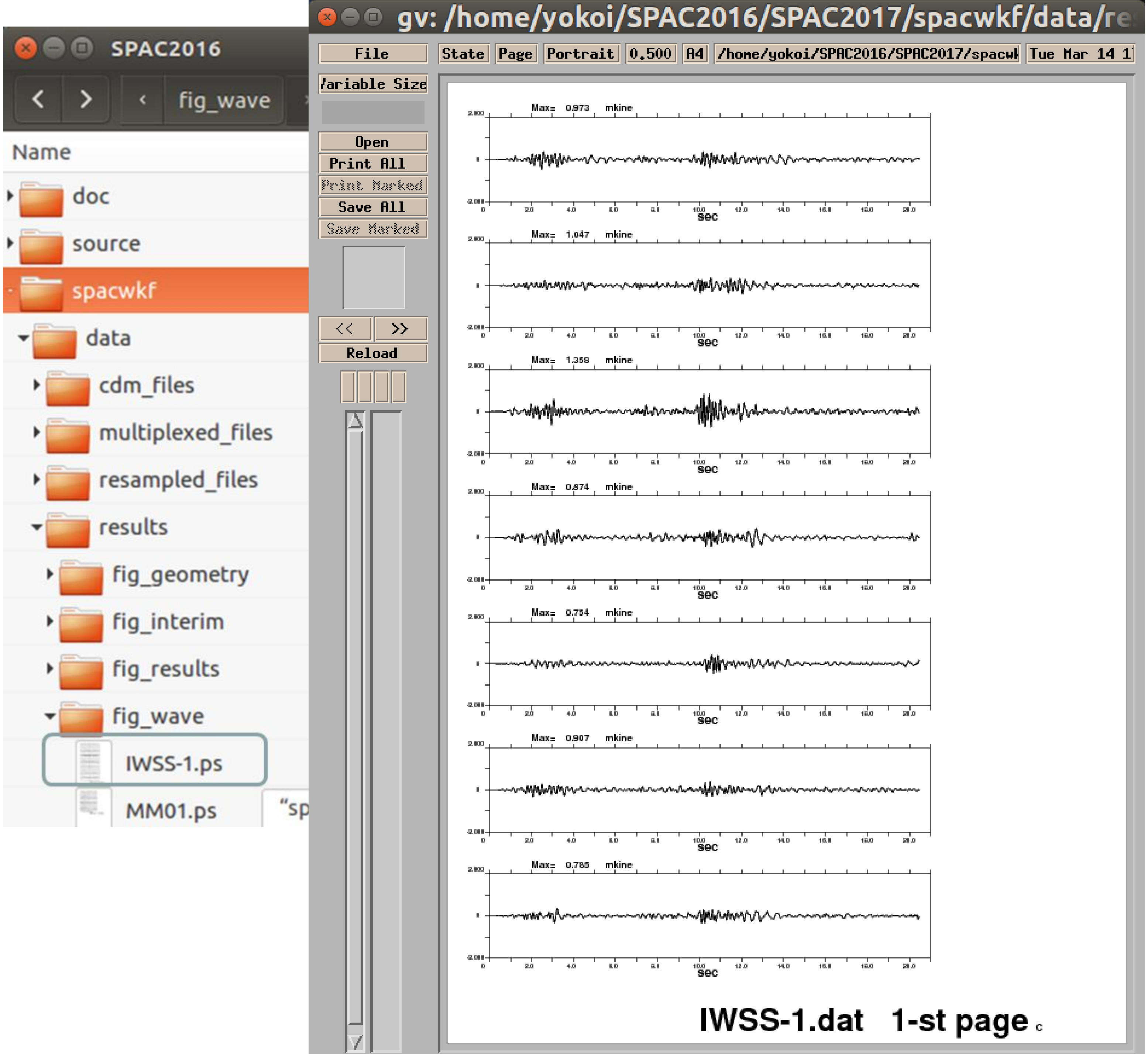
Parameter file:
./spacwkf/prm/seeblk1.prm

```
0  0.005 1.0 1.5 3     :nfilter,fl,fh,fs,nchara(=2:lowpass, =3:bandpass)
 1 10                  :n_mea,n_character
IWSS-1.dat
```

Example of execution: Output file is stored in ./spacwkf/data/results/fig_wave'

```
~SPAC2019$ .sh seeblk1.sh
 ./spacwkf/prm/seeblk1.prm
 ./spacwkf/data/resampled_files/IWSS-1.dat
          28          10
          28          20
   4.42062318E-03   4.99999989E-03
./spacwkf/data/results/fig_wave/IWSS-1.ps
~SPAC2019$
```

Multi-page postscript files are created in ./spacwkf/data/results/fig_wave



IWSS-1.dat 1-st page

1.2 Calculation of Inter station Distance and Azimuth & Plot Geometry of array

Programs used:

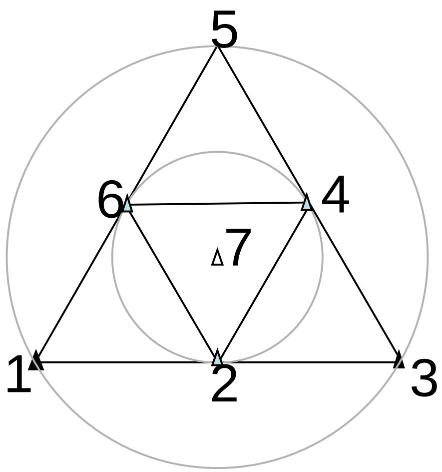
distazi.sh

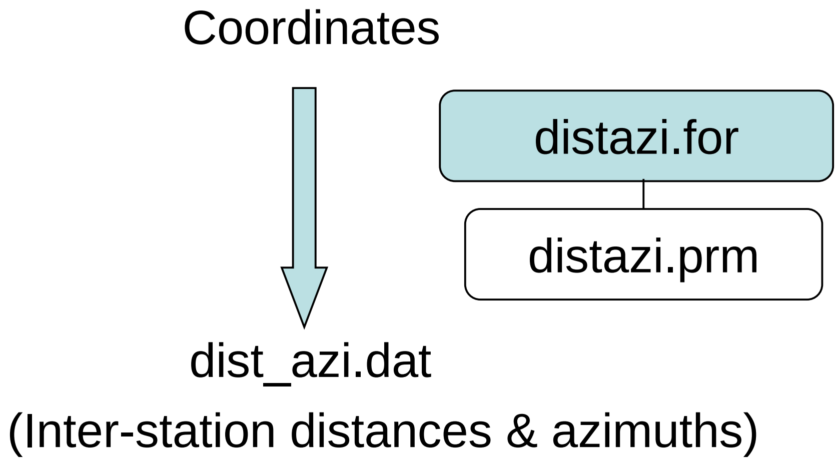
that controls:

distazi.exe + ./spacwkf/prm/distazi.prm

geometry.exe + ./spacwkf/prm/distazi.prm

A parameter file is used: temp0.dat





distazi.sh

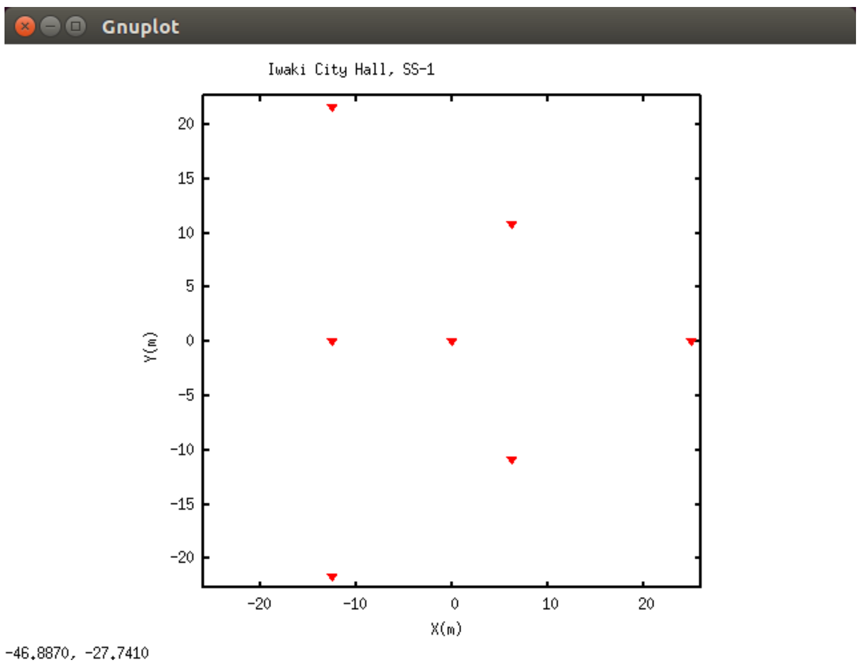
```
#!/bin/sh
rm ./spacwkf/data/results/temp0.dat
# rm ./spacwkf/data/results/distazi.dat
# rm ./spacwkf/prm/
./bin/distazi.exe
./bin/geometry.exe
gnuplot -e "
load 'geometry.plt' ;
pause -1
"
```

gnuplot is used for plotting.
If not yet installed,
sudo apt-get install gnuplot –x11

Example：SS-1 Array

distazi.prm

```
Iwaki City Hall, SS-1                       :Explanation (a45)
2                   :Type X-Y(=0), Lat-Long(=1),Dist-Angle(=2)
7   SS-1            :Number of stations, Title of figure
 25.0     90.0   1
 12.50    30.0   2
 25.0    -30.0   3
 12.50   -90.0   4
 25.00 -150.0   5
 12.50   150.0   6
  0.0      0.0   7
```



dist_azi.dat

```
Iwaki City Hall, SS-1
  7        :number of station
 id  st1  st2 distance  azimuth
  1    2    7   12.500 -150.000
  2    4    7   12.500   90.000
  3    6    7   12.500  -30.000
  4    1    2   21.651  -60.000
  5    1    6   21.651 -120.000
  6    2    3   21.651  -60.000
  7    2    4   21.651 -120.000
  8    2    6   21.651  180.000
  9    3    4   21.651  180.000
 10    4    5   21.651 -180.000
 11    4    6   21.651  120.000
 12    5    6   21.651   60.000
 13    1    7   25.000  -90.000
 14    3    7   25.000  150.000
 15    5    7   25.000   30.000
 16    1    4   37.500  -90.000
 17    2    5   37.500 -150.000
 18    3    6   37.500  150.000
 19    1    3   43.301  -60.000
 20    1    5   43.301 -120.000
 21    3    5   43.301 -180.000
```

The same image is saved simultaneously in a PostScript file in the subfolder “./spacwkf/data/results/fig_geometry/”.

distazi.prm

```
 Iwaki City Hall, SS-1                                  :Explanation (a45)
 2                        :Type X-Y(=0), Lat-Long(=1),Dist-Angle(=2)
 7   SS-1                 :Number of stations, Title of figure
   25.0     90.0    1
   12.50    30.0    2
   25.0    -30.0    3
   12.50   -90.0    4
   25.00 -150.0     5
   12.50   150.0    6
    0.0      0.0    7
```

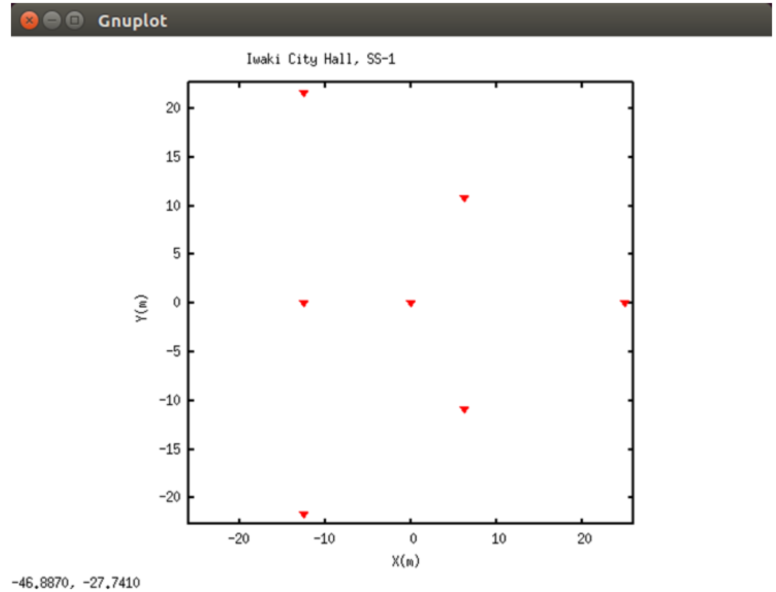
The 1st line & the 2nd parameter of the 3rd line are read and used later.

For Lat-Long (=1):
The statement for reading in distazi.for is:

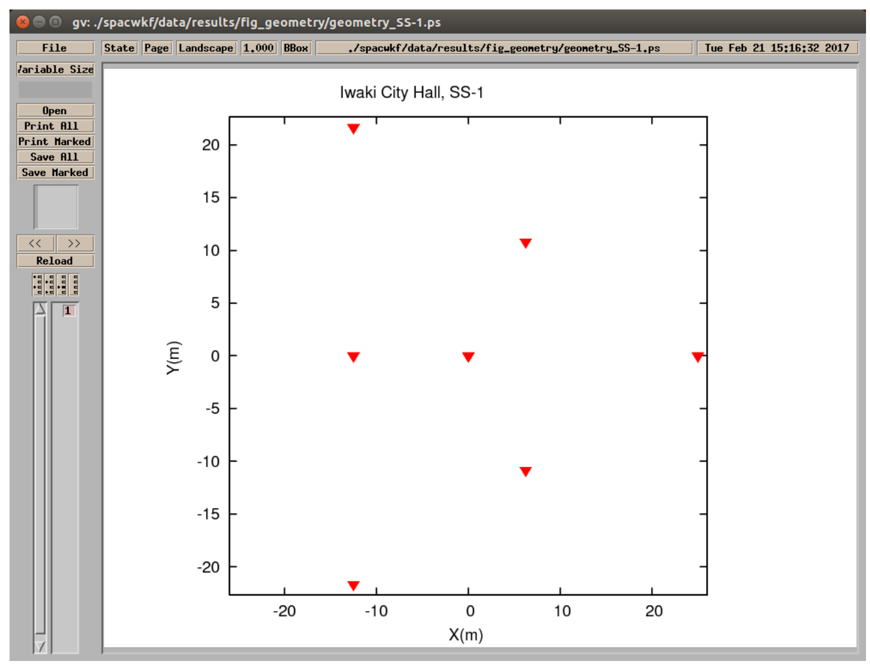
read(1,*)xla(ista),ylo(ista),id(ista)

Example of execution:

```
~SPAC2019$ sh distazi.sh
  Dist-Azimuth coordinates are selected
 X-axis toward East, Y-axis toward North.
      1   25.0    -0.0
      2    6.2    10.8
      3  -12.5    21.7
      4  -12.5    -0.0
      5  -12.5   -21.7
      6    6.3   -10.8
      7    0.0     0.0
          1          2   21.6506348      -59.9999962
...
          6          7   12.5000010      -30.0000038
 ./spacwkf/data/
 ./spacwkf/data/results/distazi.dat
   1    2    7   12.500 -150.000
...
  21    3    5   43.301 -180.000
Iwaki City Hall, SS-1
          7 SS-1
  25.0000000      -9.99999997E-07
  6.25000000       10.8253174
 -12.5000000       21.6506348
 -12.5000000      -9.99999997E-07
 -12.5000019      -21.6506348
  6.25000095      -10.8253174
  0.00000000       0.00000000

~SPAC2019$ gv &
```



Plot appears in a pop-up window



Automatically stored in a PS file, too.

Output files

./spacwkf/data/results/distazi.dat

./spacwkf/data/results/geometry.dat

./spacwkf/prm/gnuplt_script/geometry_SS-1.plt

geometry.plt

Contents of ‘geometry.plt’

load './spacwkf/prm/gnuplt_script/geometry_SS-1.plt'

./spacwkf/data/results/fig_geometry/geometry_SS-1.ps

Example: MM-array

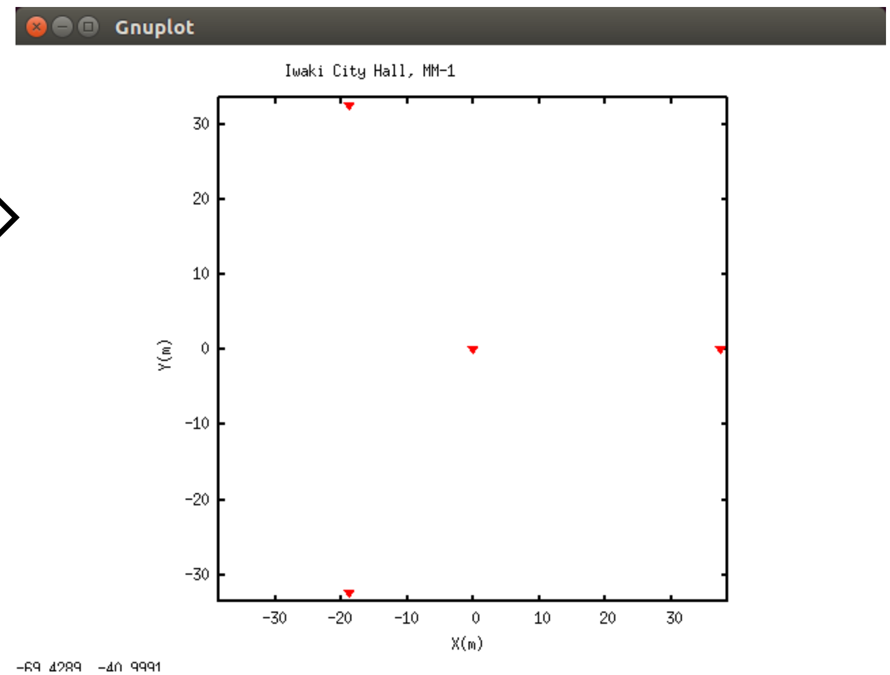
distazi.prm

```
Iwaki City Hall, MM-1                    :Explanation (a45)
2              :Type X-Y(=0), Lat-Long(=1),Dist-Angle(=2)
4   MM-1           :Number of stations, Title of figure
  37.5    90.0   1
  37.5   -30.0   2
  37.5  -150.0  3
   0.0     0.0     4
```

Then run distazi.sh

distazi.dat

```
Iwaki City Hall, MM-1
    4       :number of station
          ID          X            Y
           1   37.500000   -0.000002
           2  -18.750000   32.475952
           3  -18.750002  -32.475952
           4    0.000000    0.000000
    id  st1   st2 distance  azimuth
    1    1     4   37.500  -90.000
    2    2     4   37.500  150.000
    3    3     4   37.500   30.000
    4    1     2   64.952  -60.000
    5    1     3   64.952 -120.000
    6    2     3   64.952 -180.000
```



2. Calculation of SPAC coefficient

Program used:

zcorrel.sh

that controls

./bin/zcorrel5_3.exe + ./spacwkf/prm/zcorrel5_3.prm

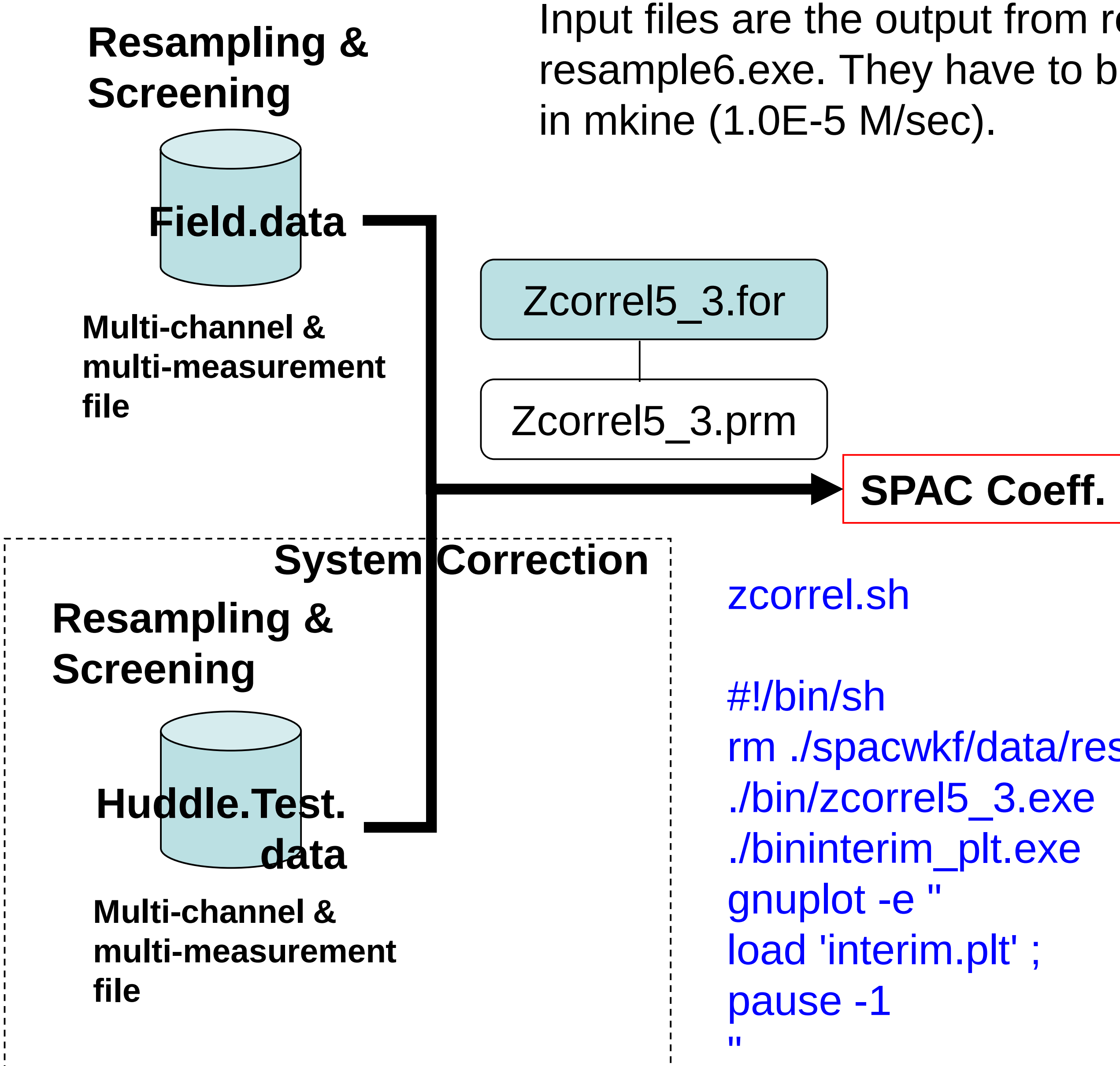
and

./bin/interim_plt.exe + ./spacwkf/prm/zcorrel5_3.prm

for plotting the results

Calculation of SPAC coefficients

Input files are the output from resample5.exe or resample6.exe. They have to be velocity records in mkine (1.0E-5 M/sec).



zcorrel.sh

```
#!/bin/sh
rm ./spacwkf/data/results/temp*.dat
./bin/zcorrel5_3.exe
./bininterim_plt.exe
gnuplot -e "
load 'interim.plt' ;
pause -1
"
```

zcorrel5_3.prm

```
Iwaki City Hall, SS-1
0.1 10.0 0.02 0.25  0 2 :fmin,fmax,dt,bw(>3.71/Td),n_huddle,nhide
HDSS-1.dat   1 1        :Huddle test file name (A12),n_coh_hud,n_pow_hud
IWSS-1.dat   1 1        :input file name (A12),n_coh,n_pow
 5
 3 27 47 67                          12.5
 9 12 16 23 24 26 34 45 46 56        21.65
 3 17 37 57                          25.0
 3 14 25 36                          37.5
 3 13 15 35                          43.3
 SPCSS1.dat             :output file name(a10)
```

1st line is used as the title of figures created

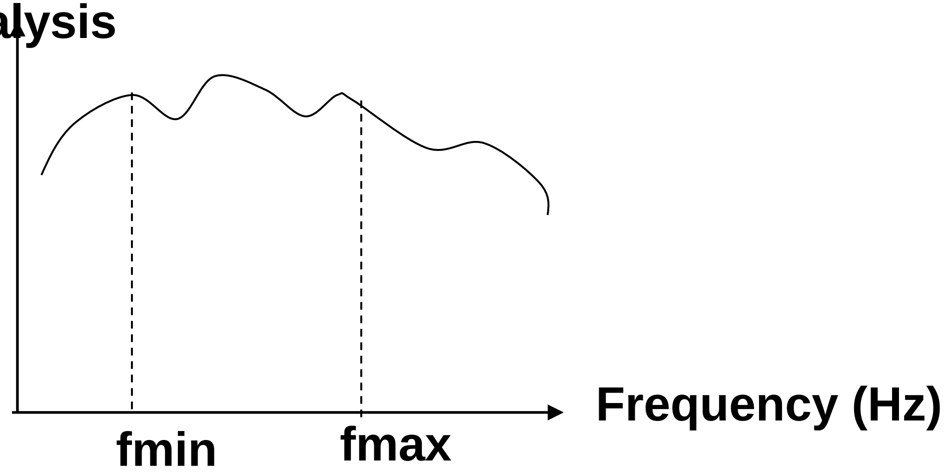
For the inter-station distance, 12.5m, 3 pairs namely stations (#2,#7),(#4,#7),(#6,#7) are averaged.

In the program nn=nn_org*nhide is conducted, where nn_org denotes length of a time block in the input data file. This means the zero padding in order to shorten the interval of frequency for the calculation of FFT.
Time window length including padded zeros is given by Td=nn_org*nhide*dt

1st line

Parameters: fmin, fmax, bw, n_huddle

Selection of frequency range for analysis



bw: bandwidth of Parzen Window that is used for smoothing the averaged cross- and auto- correlations. The bigger value of “bw” corresponds to smoother SPAC coefficient. bw=0.0 means exection without applying Parzen Window.

n_huddle: Flag for executing the system correction using the huddle test data (0= no effect, =1 conducting system correction). Correction using the huddle test data is usually not necessary when the array is composed of the seismographs of the same type.

2nd line
Parameters: Huddle test file name (A12),n_coh_hud,n_pow_hud

Resampled & screened Multi-channel & multi-measurement file of the huddle test data

***huddle*.dat**

Flag for outputing the coherence file of the huddle test data
0= no effect
1= output

***huddle*_coh.dat**

Flag for outputing the power spectra file of the huddle test data
0= no effect
1= output

***huddle*_psp.dat**

This line is read but not used in the analysis if n_huddle in the 1st line is “0”.

3rd line
Parameters: input file name (A12),n_coh,n_pow

Resampled & screened Multi-channel & multi-measurement file of the huddle test data

***Input*.dat**

Flag for outputing the coherence file of the huddle test data
0= no effect
1= output

***Input*_cor_coh.dat**

Flag for outputing the power spectra file of the huddle test data
0= no effect
1= output

***Input*_cor_psp.dat**

The coherence and the power spectra of the field data without the system correction are output as default.

***Input*_coh.dat**
***Input*_psp.dat**

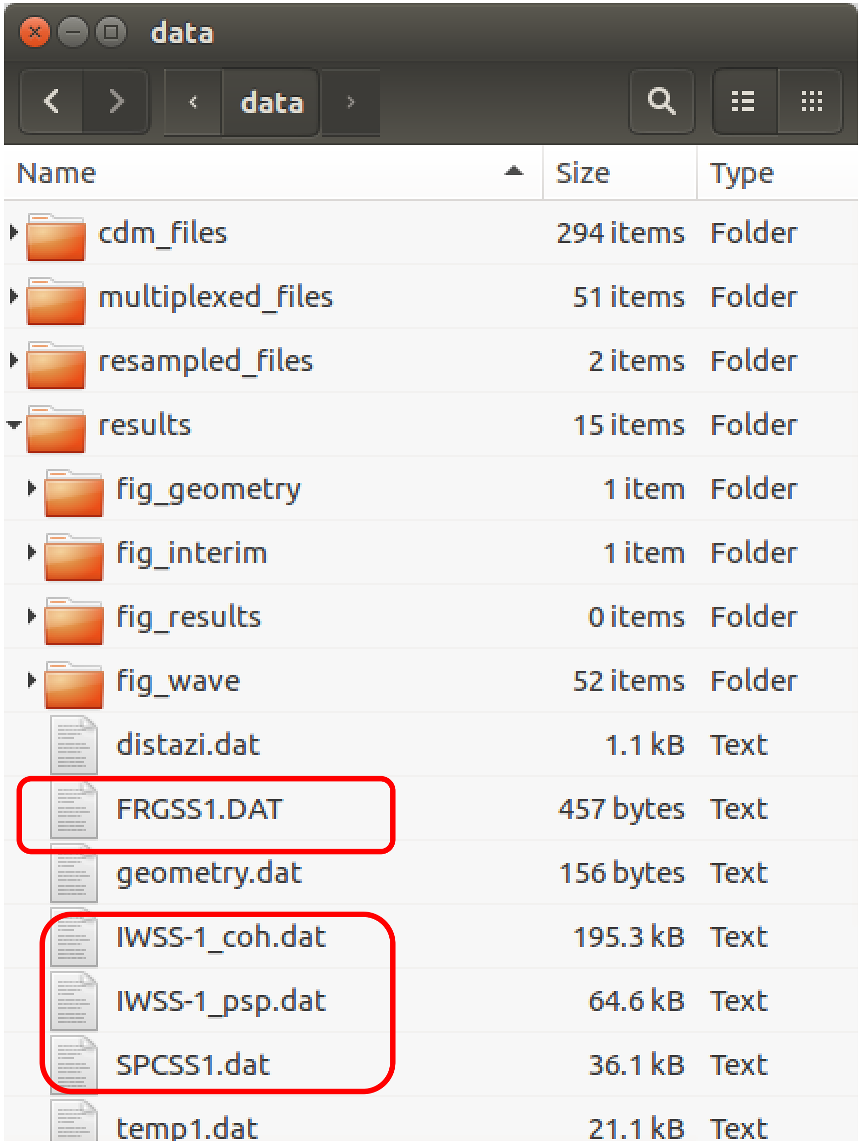
Example of execution:

zcorrel5_3.exe

```
~SPAC2019$ sh zcorrel.sh
Program zcorrel5_3.for
./spacwkf/prm/zcorrel5_3.prm
 Huddle Test Skipped.
nn=  2048
         28          10
         28          20
 power and cross spectra for field data calculated.
Block Averaging has been done.
./spacwkf/data/results/IWSS-1_psp.dat
Output:./spacwkf/data/results/IWSS-1_psp.dat
./spacwkf/data/results/IWSS-1_coh.dat
./spacwkf/data/results/SPCSS1.dat
```

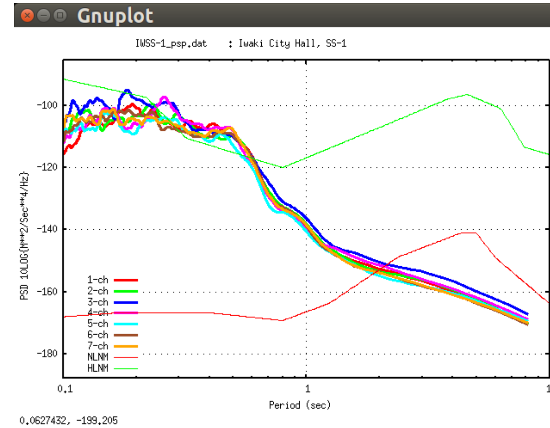
interim_plt.exe

Look the next page

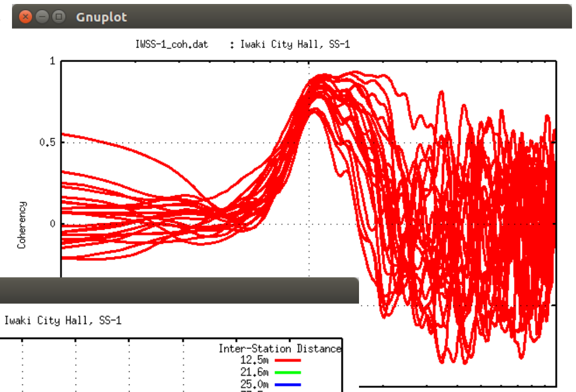


interim_plt.exe

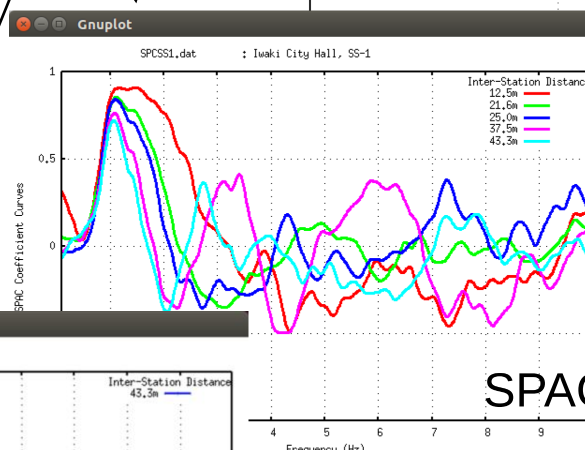
```
Iwaki City Hall, SS-1
Iwaki City Hall, SS-1
Hit return to continue
Hit return to continue
Hit return to continue
Hit return to continue
Hit return to continue
Hit return to continue
Hit return to continue

~SPAC2019$
```



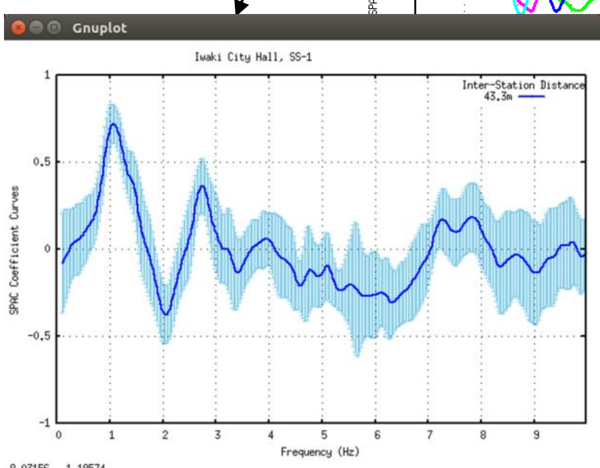
Power spectra



All Coherence functions



SPAC coefficients

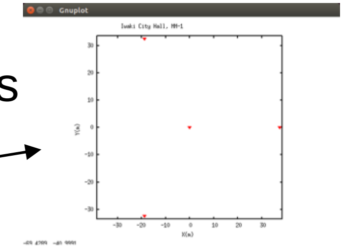


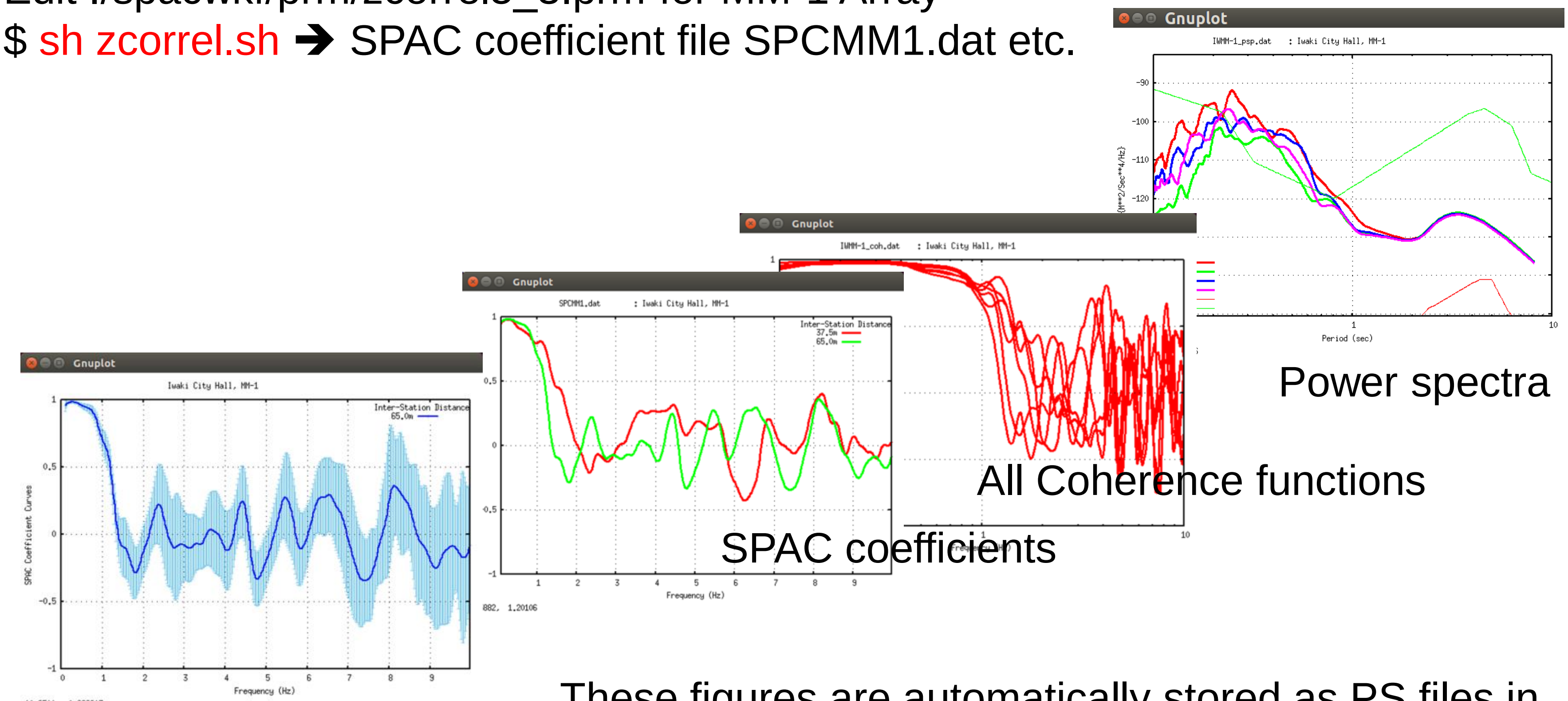
Individual SPAC function with error bar

These figures are automatically stored as PS files in ./spacwkf/data/results/fig_interim

Procedure for MM-1 Array

+ Edit ./spacwkf/prm/resample6.prm for MM-1 Array
+ $ sh resample5.sh ➔ resampled & screened file IWMM-1.dat
+ Edit ./spacwkf/prm/seewav6.prm for MM-1 Array
+ $ sh seewav6.sh ➔ Waveform plot files MM??.ps
+ Edit ./spacwkf/prm/distazi.prm for MM-1 Array
+ $ sh distazi.sh ➔geometry plot file geometry_MM-1.ps
+ Edit ./spacwkf/prm/zcorrel5_3.prm for MM-1 Array
+ $ sh zcorrel.sh ➔ SPAC coefficient file SPCMM1.dat etc.



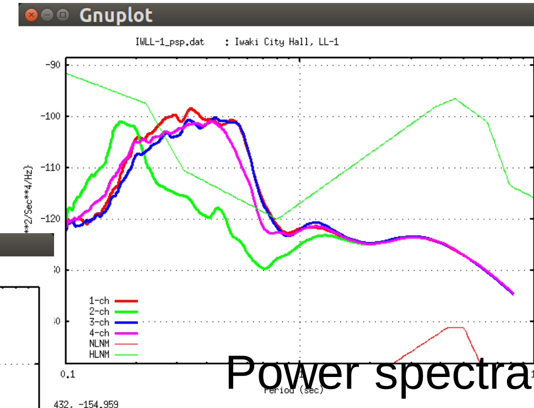


Individual SPAC function with error bar

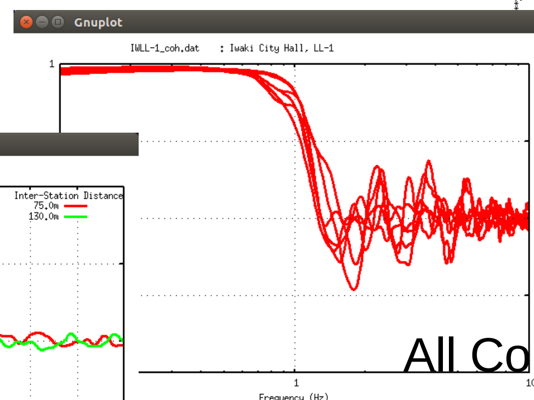
These figures are automatically stored as PS files in ./spacwkf/data/results/fig_interim

Procedure for LL-1 Array

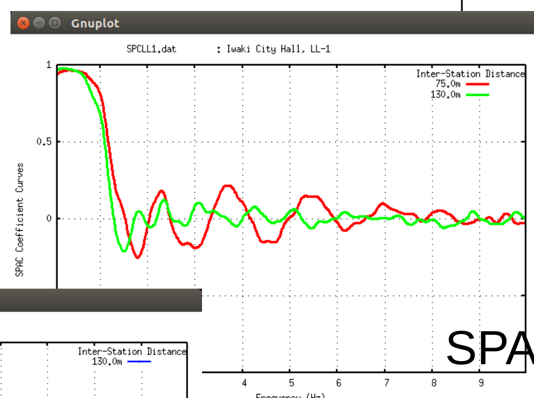
+ Edit ./spacwkf/prm/resample6.prm for LL-1 Array
+ $ sh resample6.sh ➔ resampled & screened file IWLL-1.dat
+ Edit ./spacwkf/prm/seewav6.prm for LL-1 Array
+ $ sh seewav6.sh ➔ Waveform plot files LL??.ps
+ Edit ./spacwkf/prm/distazi.prm for LL-1 Array
+ $ sh distazi.sh ➔geometry plot file geometry_LL-1.ps
+ Edit ./spacwkf/prm/zcorrel5_3.prm for LL-1 Array
+ $ sh zcorrel.sh ➔ SPAC coefficient file SPCLL1.dat etc.



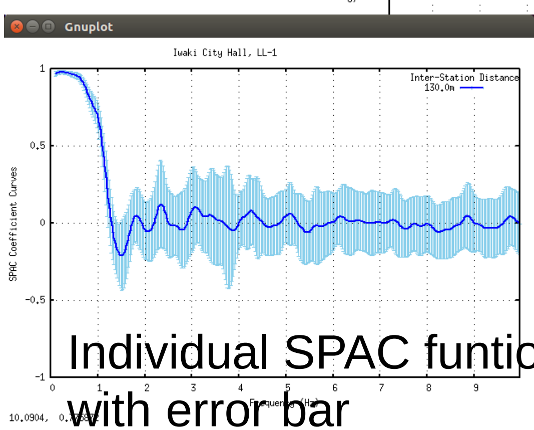
Power spectra



All Coherence functions



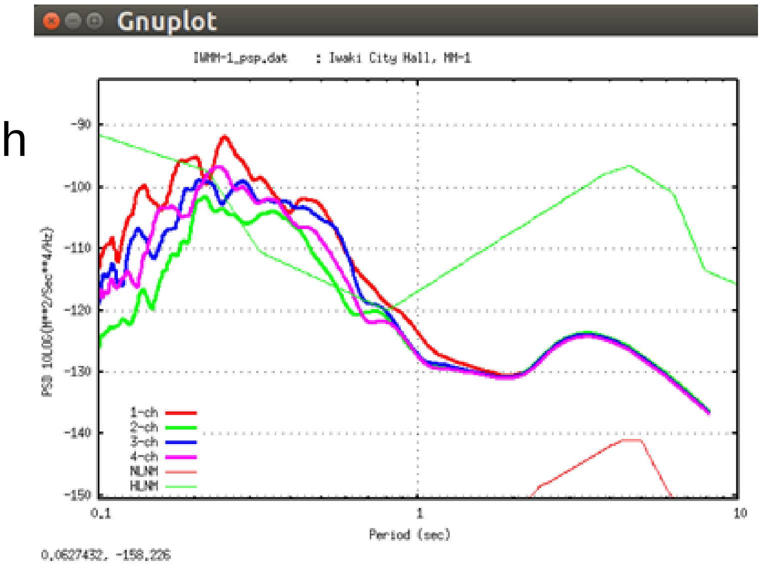
SPAC coefficients



Individual SPAC funtion
with error bar

These figures are automatically stored as PS files in
./spacwkf/data/results/fig_interim

Power Spectra



10*$\log_{10}$ (Fourier Coefficient *Td* omg**2) of each frequency is plotted in PostScript file
spacwkf/data/results/fig_interim/IWMM-1_psp.ps
in the unit 10*Log10(M**2/Sec**4/Hz)
Also output into
spacwkf/data/results/IWMM-1_psp.dat
that is CSV format file.

IWMM-1_psp.dat

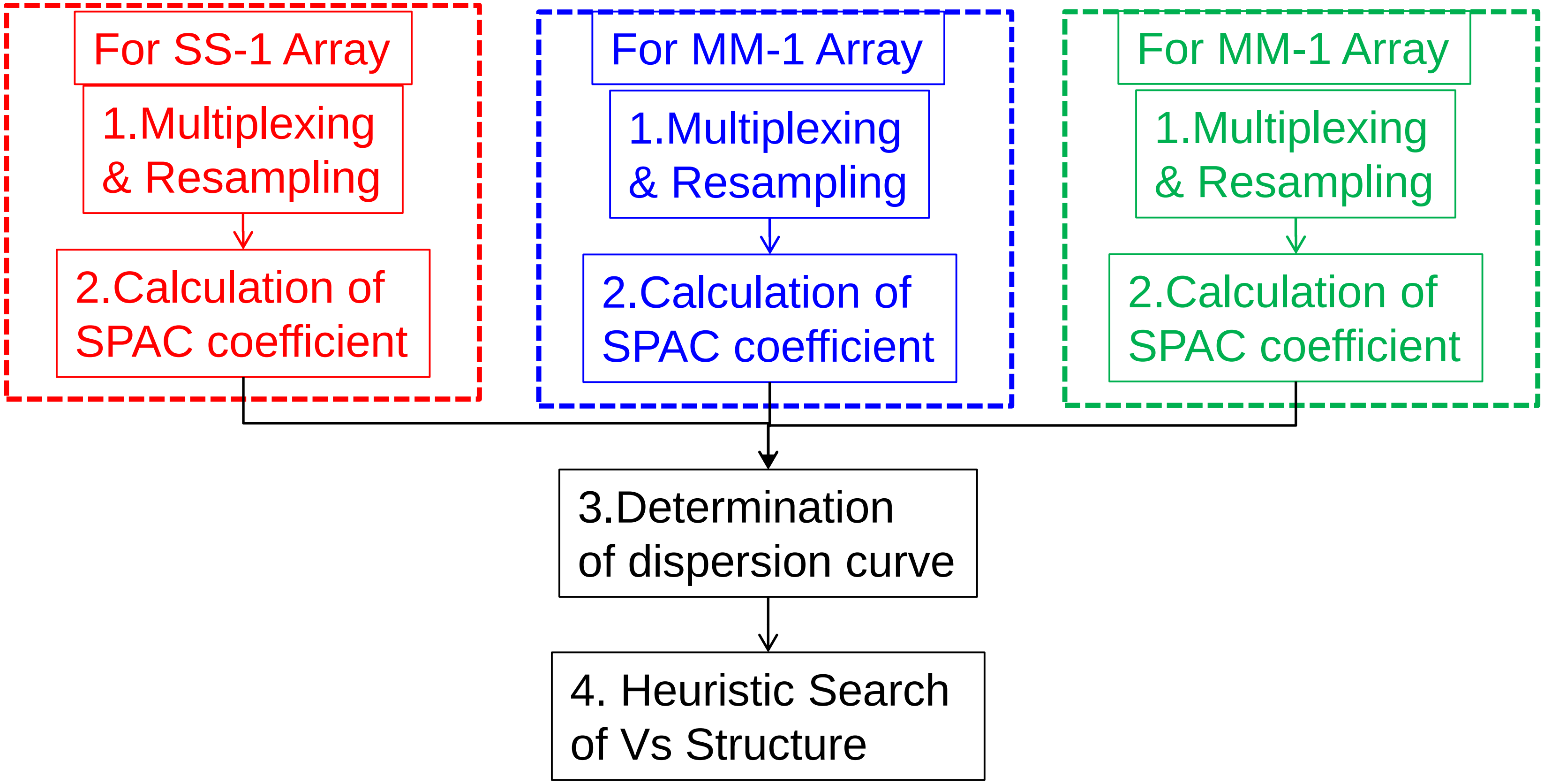
# Period	1-ch	2-ch	3-ch	4-ch Power Spc	1-ch	2-ch	3-ch	4-ch S.D.
0.100	-122.523	-121.038	-123.098	-122.525	-123.525	-121.061	-123.484	-122.815
0.100	-121.617	-120.174	-122.290	-121.659	-122.703	-120.218	-122.836	-122.091
0.101	-120.964	-119.546	-121.741	-121.039	-122.073	-119.536	-122.445	-121.599
…								

10*Log10 (Fourier Coefficient * Td * omg**2)

10*LOG10(sqrt(variance of Fourier Coefficient)*Td*omg**2)

where omg=2π/period
When Velocity Fourier Amplitude Spectral Density is required, type
sh fourier.sh
in the folder SPAC2017. fourier_plt.exe convert the above mentioned psp to Fourier Amplitude Spectra.

Example: Up to this slide the red part of the below shown diagram has been performed for SS-1 Array (7 points). The next step is to do the same procedure for MM-1 Array (4 points), i.e., the blue part of the diagram below. Then, the green part.



SPCSS1.dat, SPCMM1.dat, SPCLL1.dat are used in the next step of analysis.

3. Determination of Dispersion Curve

Programs used:

cf_panel1.sh

that controls

./bin/cf_panel1.exe + cf_panel1.prm

for fitting to Bessel function, plotting SPAC coefficient and determination of dispersion curve using the imaging technique.

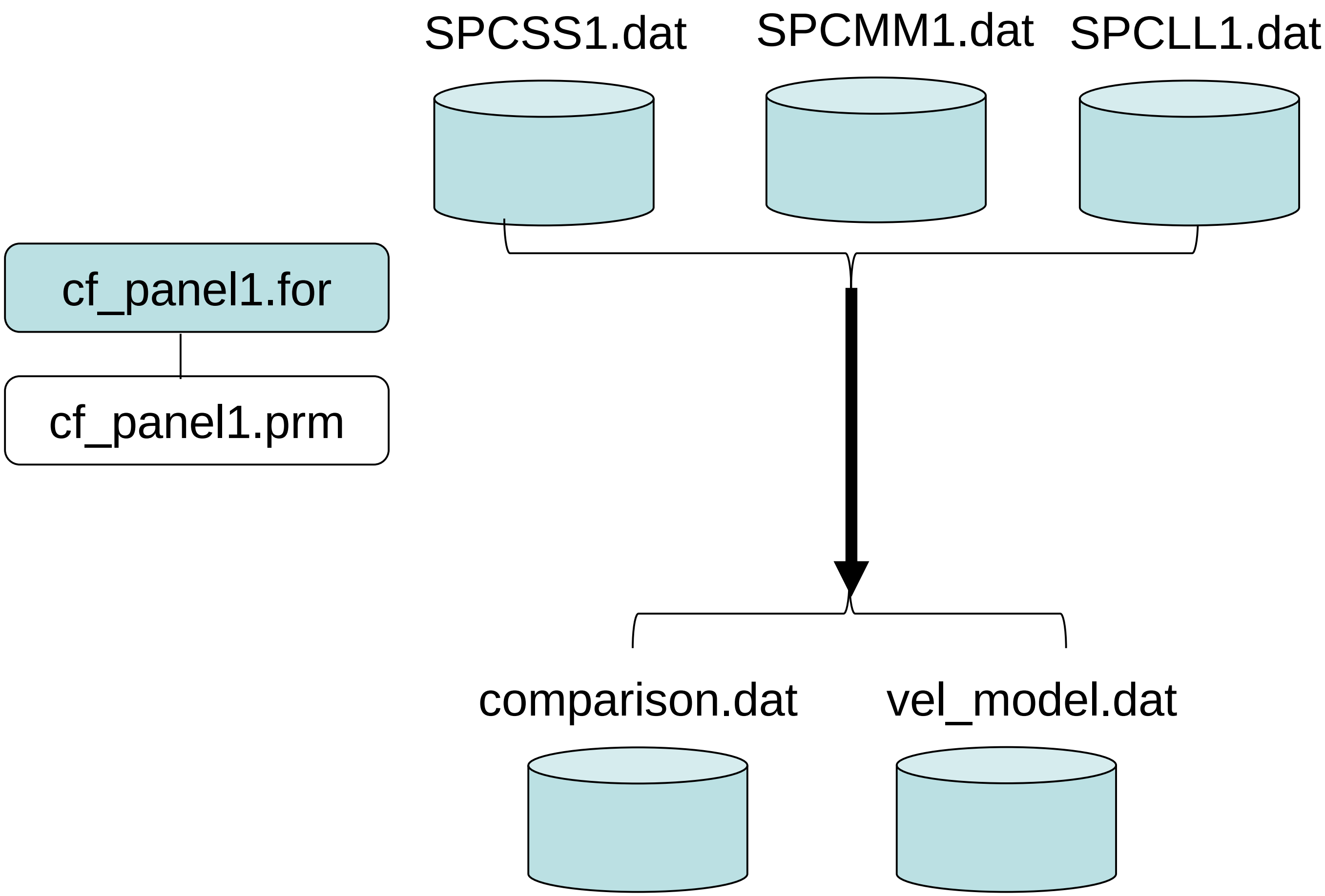
For plotting

vel_model.sh

that controls

./bin/vel_model_plt.exe + vel_model_plt.prm

All input files must have the same df=1/Td



The same file name is used for every case

Determination of dispersion curve

1st step:

+ The fif-th order polynomial that approximates the inverse function of $J_o(x)$

$$y=-6.0803x^5+9.2477x^4-3.9322x^3+0.1815x^2-1.7079x+2.4121$$

is used to calculate the tentative values of the phase velocity $c(r,\omega)$, where $y=kr=r\omega/c(r,\omega)$, $x=\rho(r,\omega)$.

+ The wave lengths are estimated from the observed SPAC coefficients through this tentative phase velocity.

+ The frequency range for analysis is narrowed below the frequency that corresponds to the spacial Nyquist wave length, i.e., 2 times the minimum interstation distance.

+ The frequency range for analysis is again narrowed above the frequency of the maximum value of SAC coefficient curves.

2nd step:

+ Imaging technique is applied to determine the phase velocity of each frequency. The misfit function is set for the SPAC coefficient curves.

Example:

cf_panel1.prm

```
Iwaki city Hall (test data)    2017 version        :graph title(a50)
 0  1  0 40.96  0.1     :n_message,n_comparison(0=no, 1=yes),nfskip(0=no, 1=yes),Td, aps
 0.10  5.0  100.0  1000.0 :fmin,fmax,vmin,vmax that specify the range of analysis
3                        :file number
SPCSS1.dat               :file name (a12)
    5                   :Number of Radius
  12.5  1.0               :Radius(m),Min and Max of the frequency(Hz) for fitting
  21.6  1.0
  25.0  1.0
  37.5  1.0
  43.3  1.0
SPCMM1.dat               :file name (a12)
    2                   :Number of Radius
  37.5                    :Radius(m),Min and Max of the frequency(Hz) for fitting
  65.0
SPCLL1.dat               :file name (a12)
    2                   :Number of Radius
  75.0                    :Radius(m),Min and Max of the frequency(Hz) for fitting
 130.0
```

SS-1-array

MM-1-array

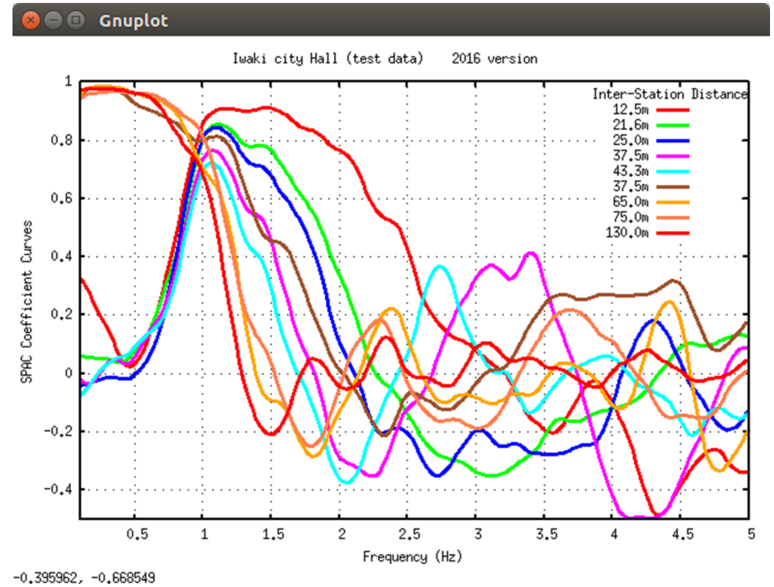
LL-1-array

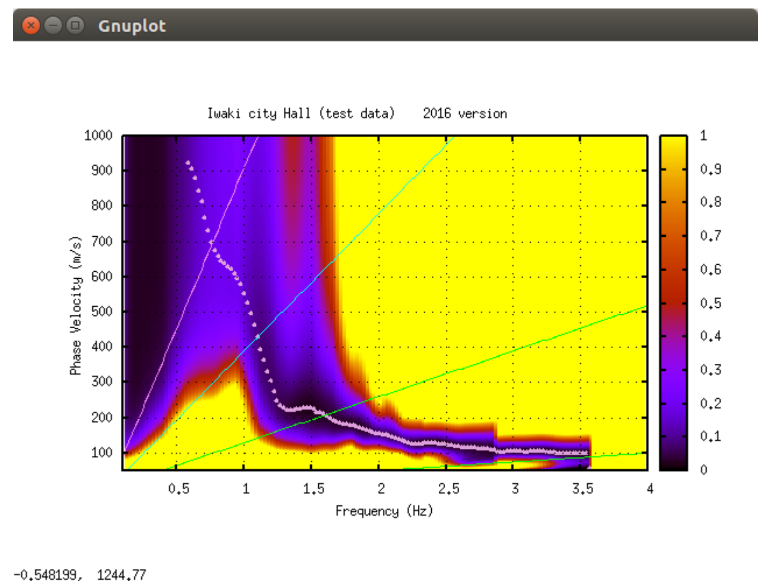
→ The list of inter-station distances must be coincident to the contents of the SPAC coefficient's file.

The minimum and maximum frequency for fitting can be specified, if not f_{min} and f_{max} in the 3rd line will be used in place of them.

Example of execution:

```
$ sh cf_panel1.sh
 n_message=  0 n_comparison=  1 nfskip=  0   0.200E+00
 Range of Analysis: from     0.10(Hz) to     5.00(Hz)
                and from    50.00(m/s) to  1000.00(m/s)
 Number of files=   3 for SPAC Coef. curves.
 1  SPCSS1.dat
 2  SPCMM1.dat
 3  SPCLL1.dat
 No. Distance min_fr  max_fr
              m      Hz      Hz
      1   12.50    1.47    3.56
      2   21.60    1.42    2.86
      3   25.00    1.10    2.32
      4   37.50    1.07    2.03
      5   43.30    1.05    1.93
      6   37.50    1.10    2.34
      7   65.00    0.24    1.81
      8   75.00    0.34    1.81
      9  130.00    0.17    1.51
 Range of analysis: fmin=  0.561999977      fmax=  3.56399989
 ./spacwkf/prm/gnuplt_script/SPAC_coef.plt
 ./spacwkf/prm/gnuplt_script/cf_panel1.plt
Hit return to continue
```





These figures are automatically stored as PS files in
 ./spacwkf/data/results/fig_results

Example of Output

vel_model.dat for the determined dispersion curve

```
#  freq.(Hz) Vel.(km/s)
     0.586      0.925      0.004
     0.610      0.907      0.004
     0.635      0.881      0.004
     0.659      0.845      0.004
     0.684      0.813      0.005
…
```

Standard Deviation

Comparison.dat

for checking the glade of fitting

```
…
6   37.500     0.071     -0.180     0.199
f=    1.953       Hz   160.00        =c
      r    SPAC_Coef.    Cal_Bessel    S.D.
     0.000               1.000
1   12.500     0.777      0.783     0.076
2   21.600     0.388      0.423     0.091
3   25.000     0.176      0.272     0.205
4   37.500    -0.241     -0.215     0.185
6   37.500     0.047     -0.215     0.197
f=    1.977       Hz   157.00        =c
      r    SPAC_Coef.    Cal_Bessel    S.D.
…
```

S.D.: Standard deviation of the observed SPAC coefficient

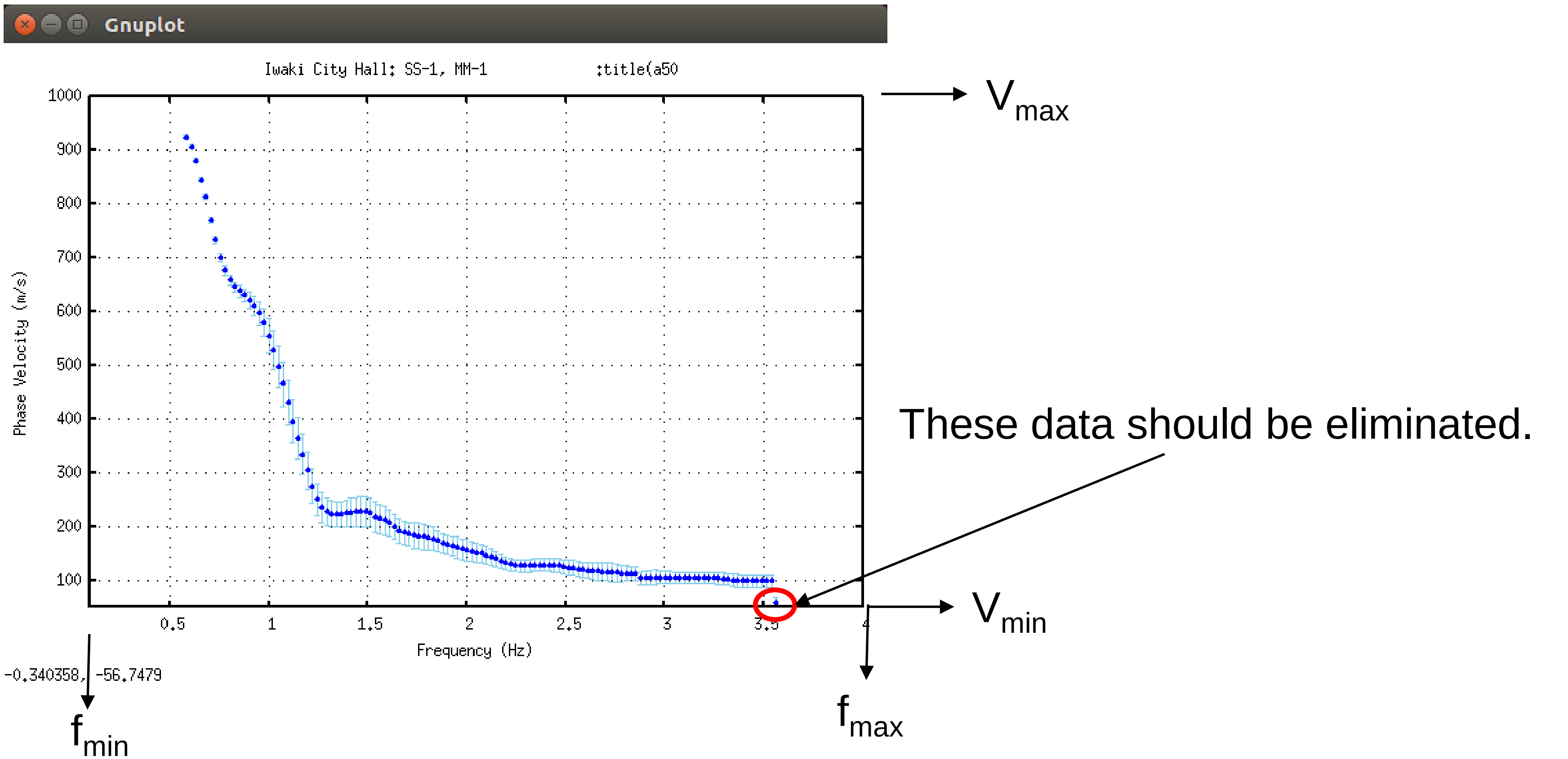
Cal_Bessel: SPAC coefficient calculated using determined phse velocity

SPAC_Coef.: Observed SPAC coefficient

r: Inter-station distance

Plot the determined dispersion curve:
Execution: After editing ./spacwkf/prm/vel_model_plt.prm
for the graph title, [fmin,fmax] and [vmin, vmax]

~SPAC2019$ sh vel_model.sh
./spacwkf/prm/gnuplt_script/vel_model.plt



This figure is automatically stored as PS files in
./spacwkf/data/results/fig_results

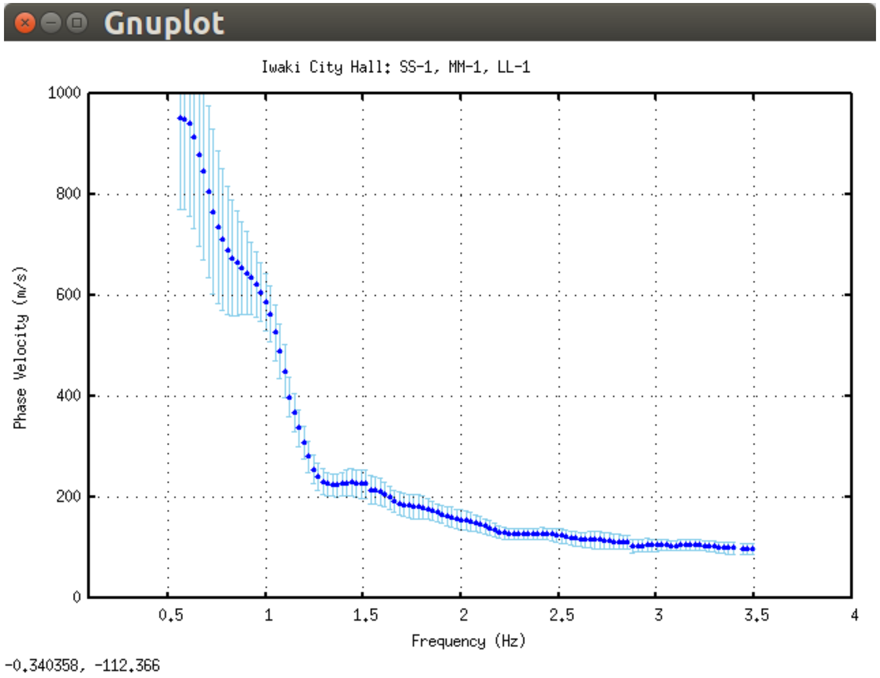
./spacwkf/prm/vel_model_plt.prm

```
Iwaki City Hall: SS-1, MM-1, LL-1                :title(a50)
  0.1    4.0                      : fmin,fmax
  0.  1000.                       : vmin,vmax
 1                                : n_mod
```

1st line: title of phase velocity plot
2nd line: fmin and fmax for plotting
3rd line: vmin and vmax for plotting
4th line: n_mod
=1: wavelength/3 – Vs will be plotted in Vs structure plot
otherwise not plotted.

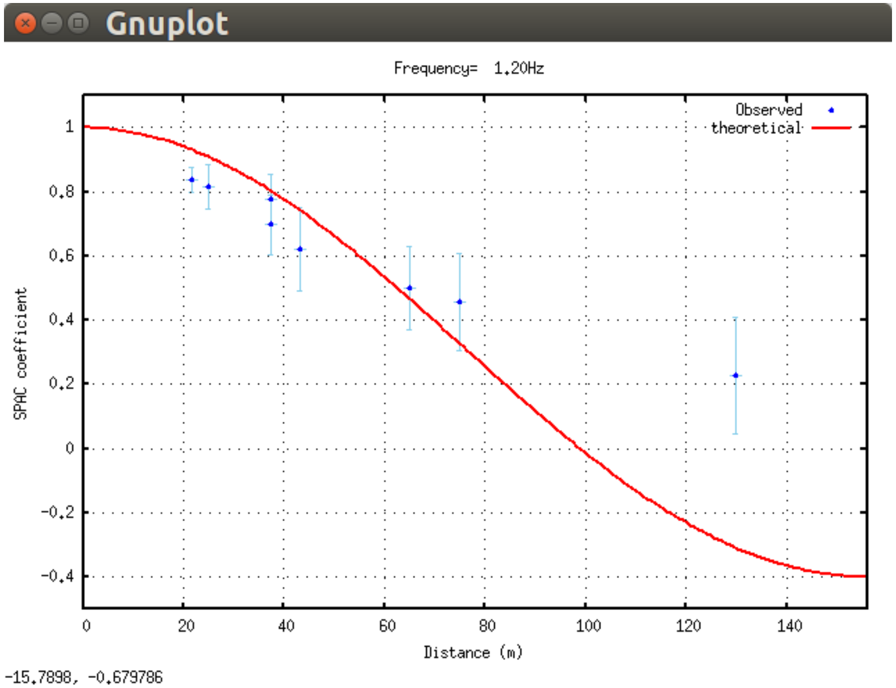
File: vel_model.dat

- [fmin, fmax] described in vel_model_plt.prm and used by vel_model_plt.exe and cf_panel.exe are not reflected to the contents of the file vel_model.dat.
- It is recommendable to edit vel_model.dat manually to select the frequency range used in the next step.
- Don’t leave a blank line at the end of the file vel_model.dat after editing.
- Run vel_model_plt.exe after editing it.➔ *sh .vel_model.sh*



Plot the SPAC coefficient curve fitting:

Execution:
$ sh comparison.sh
…
Type in the frequency that you want to select.
…
./spacwkf/prm/gnuplt_script/comparison.plt



Simultaneously the same imge is saved in a PostScript file:
./spacwkf/data/results/fig_results/comparison.ps

4. Heuristic Search of Vs Structure

Programs used:

inversion.sh

that controls

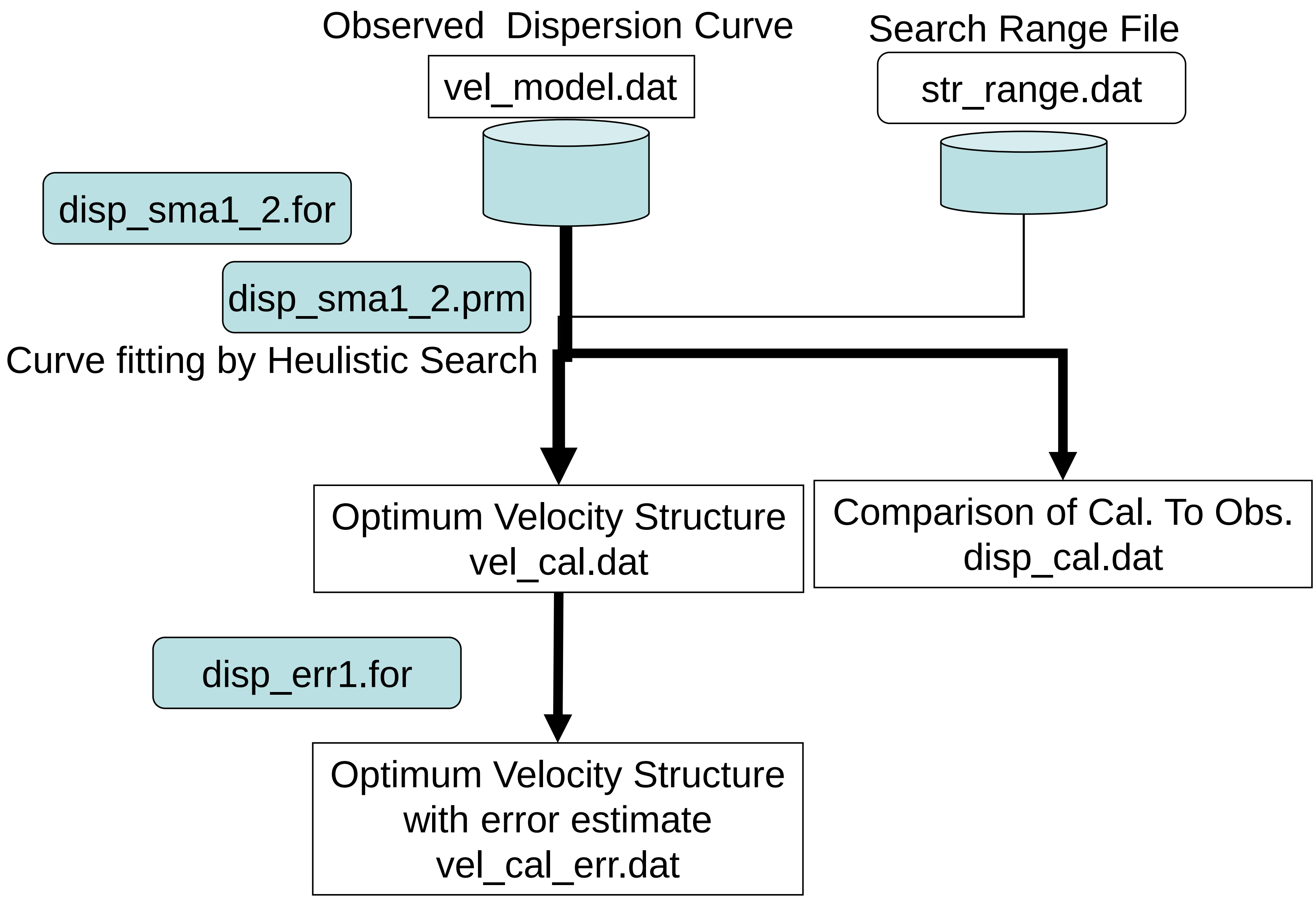
./bin/disp_sma1_2.exe + ./spacwkf/prm/disp_sma1_2.prm

./bin/disp_err1.exe

results.sh

that controls

./bin/results_plt.exe


Observed Dispersion Curve
vel_model.dat
Search Range File
str_range.dat
disp_sma1_2.for
disp_sma1_2.prm
Curve fitting by Heulistic Search
Optimum Velocity Structure
vel_cal.dat
Comparison of Cal. To Obs.
disp_cal.dat
disp_err1.for
Optimum Velocity Structure
with error estimate
vel_cal_err.dat

inversion.sh

```
#!/bin/sh
rm ./spacwkf/data/results/temp*.dat
./bin/disp_sma1_2.exe
./bin/disp_err1.exe
```

results.sh

```
#!/bin/sh
./bin/results_plt.exe
gnuplot -e "load 'results.plt' ; pause -1"
```

disp_sma1_2.for

Combination of the Down Hill Simplex Method (Nelder & Mead (1965)) and the Very Fast Simulated Annealing method (Ingber, 1989).

DHSM: Down Hill Simplex Method (Nelder & Mead (1965))
An efficient algorithm to find "local minimum".
Faster than Geiger's method. Partial derivatives are not necessary.
Result is controlled by given initial values and easily captured by local minimum.
Example of application to the microtremor array: Ohori et al(2002)

VFSA: Very Fast Simulated Annealing method (Ingber, 1989)
One of the heuristic search methods.
Analogy of cooling and crystallization process of metals.
Results can escape from local minimum and can get global minimum with some probability.
Time consuming due to the probabilistic search for each parameter.
Example of application to the microtremor array & appropriate values of parameters for this purpose: Yamanaka (2004)

Example: disp_sma1_2.prm

```
 1 1. 0.6 1.3 10000  5  :idum,t0,a,c,ntemp,j0
 0.002                  :eps0
   1    1            :n_roh,n_vp
   1    0    1        :ini_flg,ndsp_flg,n_err
   0    1            :kflg,jflg
   0    0            :n_vs,n_th
str_range.dat         :File name for the initial velocity model (a25).
vel_model.dat          :File name for the obseved dispersion relation (a25).
vel_cal.dat          :File name for the estimated velocity structure (a25)
disp_cal.dat          :File name for the calculated dispersion relation (a25)
```

Control parameter for the simulated annealing method

- idum :Random seed (integer): As the result may depend on the initial velocity model given by random number, it is strongly recommended for users to apply this program several times with various values of random seed and to grasp the scatter of result.
- t0, a,c :Initial Temperature, Coefficients for T=T0*exp(-c*k**a), where k is iteration number. T0=1.0, a=0.6 & c=1.3 are of the fastest schedule
- ntemp :Maximum number of temparature change
- j0 :Number of iteration for each temperature

threshold for conversion

- eps0 : threshold of misfit function

flags for roh and vp

- n_vp : 1=by Ludwig et al(1970): vp=1.11*vs+1.29; 0=fixed to the initial values
- n_roh : 1=by Kitzunezaki et al(1990): roh=1.2475+0.399*vp-0.026*vp**2; 0=fixed to the initial values

flags for output to Display

- ini_flg : Initial Velocity Structure Model. 1=yes
- ndsp_flg : Observed Dispersion Relation . 1=yes
- n_err : Error at each iteration, 1=yes
- kflg : Missfit at each temp. change, 1=yes
- jflg : Missfit at each itration with the same temp. 1=yes
- n_vs : Vs value (n_vs=layer number, 0=no output)
- n_th : Thickness value (n_th=layer number, 0=no output)
- n_err :

Example: str_range.dat 🡨 Initial Search Range

```
Iwaki City, S-M-L 22/12/2012  :Model(a30)
        6                     :IL(I5),Layer Number
1.9   1.5    0.005  0.030  0.10 0.12 :density,Vp,hmin,hmax,vmin,vmax
1.9   1.5    0.005  0.030  0.10 0.20 :density,Vp,hmin,hmax,vmin,vmax
1.9   1.5    0.005  0.030  0.20 0.40 :density,Vp,hmin,hmax,vmin,vmax
1.9   1.5    0.005  0.300  0.40 0.70 :density,Vp,hmin,hmax,vmin,vmax
1.9   1.5    0.005  0.300  0.70 1.00 :density,Vp,hmin,hmax,vmin,vmax
2.0   1.70   998.0 999.0   1.00 3.20
```

(hmin, hmax) : Search range for layer thickness.
(vmin, vmax) : Search range for Vs (Shear Wave Velocity)

Example of execution(1): inversion.sh

```
$ sh inversion.sh
+-----------------------------------------------------+
+                                                     +
+                    Disp_sma1                        +
+                                                     +
+ Program to obtain the optimum undeground velocity +
+ structure for the given dispersion relation of    +
+ Rayleigh wave.                                    +
+                                                     +
+ The used method is a combination of the down hill +
+ simplex method (Nelder & Mead (1965)) and the     +
+ very fast simulated annealing method (Ingber      +
+ (1989)).                                          +
+                                                     +
+ The subroutine DSPRAY and DSPMRX published in     +
+ "Seismological Algorithm" are used directly.      +
+ AMOEBA and AMOTRY published in "Numerical Recipe" +
+ are also used, but with significant modification  +
+ for the adaptation with the very fast simulated   +
+ annealing method.                                 +
+                                                     +
+ By the combination with the down hill simplex     +
+ method, the very fast simulated annealing method  +
+ is gotten much faster.                            +
+                                                     +
+                                       July 6, 2005+
+    CopyRight by Toshiaki Yokoi, IISEE, BRI, Japan.+
+-----------------------------------------------------+
./spacwkf/data/results/progress.dat
./spacwkf/prm/disp_sma1_2.prm
./spacwkf/prm/str_range.dat
 Initial values randomly produced
```

Opening

Interim output

Input files

Example of execution(2):

```
 Range of random fluctuation
Thicknes Density       Vp       Vs    Thickness          Vs
   0.018    1.756    1.403    0.102    0.005    0.030    0.100    0.120
   0.026    1.770    1.445    0.139    0.005    0.030    0.100    0.200
   0.023    1.849    1.696    0.365    0.005    0.030    0.200    0.400
   0.120    1.952    2.036    0.672    0.005    0.300    0.400    0.700
   0.275    1.993    2.176    0.798    0.005    0.300    0.700    1.000
 999.000    2.356    3.645    2.121  998.000  999.000    1.000    3.200
 ./spacwkf/data/results/vel_model.dat
    5    0.0086920802
   10    0.0056342622
   15    0.0056342622
...
  855    0.0019967253
 ./spacwkf/data/results/vel_cal.dat
       Thicknes(Km) Density(g/cm^3)     Vp(Km/sec)     Vs(Km/sec)
    1       0.021193       1.755522       1.401171       0.100154
    2       0.009001       1.778089       1.470751       0.162839
    3       0.005679       1.837834       1.658847       0.332294
    4       0.297426       1.949227       2.026254       0.663292
    5       0.161642       1.997086       2.191665       0.812311
    6     999.000000       2.115754       2.625133       1.202823
 ./spacwkf/data/results/disp_cal.dat
 ./spacwkf/data/results/err_estm.dat
```

Initial model selected using random number with in the given initial search range

Input file (Dispersion Curve)

Iterations (Number , Misfit func.)

Output file

Optimum Solution

Output files

Output-1

progress.dat: Structure Models of every j0 iterations are stored.

err_estm.dat: Data for error estimation (Next Step) are stored.

Output-2

vel_cal.dat: Optimum Solution

	Thicknes(Km)	Density(g/cm^3)	Vp(Km/sec)	Vs(Km/sec)
1	0.021193	1.755522	1.401171	0.100154
2	0.009001	1.778089	1.470751	0.162839
3	0.005679	1.837834	1.658847	0.332294
4	0.297426	1.949227	2.026254	0.663292
5	0.161642	1.997086	2.191665	0.812311
6	999.000000	2.115754	2.625133	1.202823

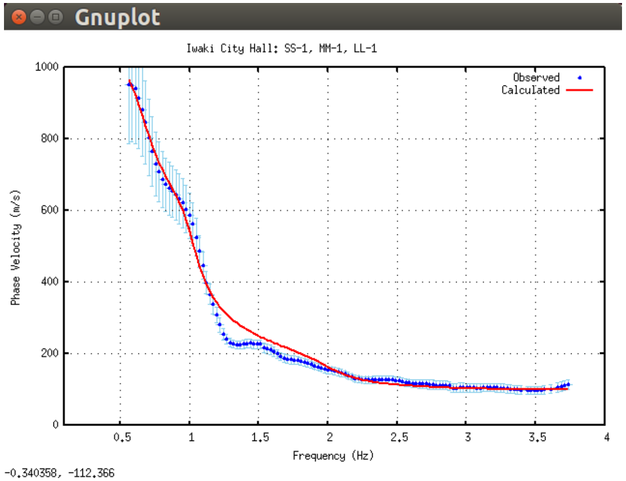
disp_cal.dat: Observed & Calculated Dispersion Curves

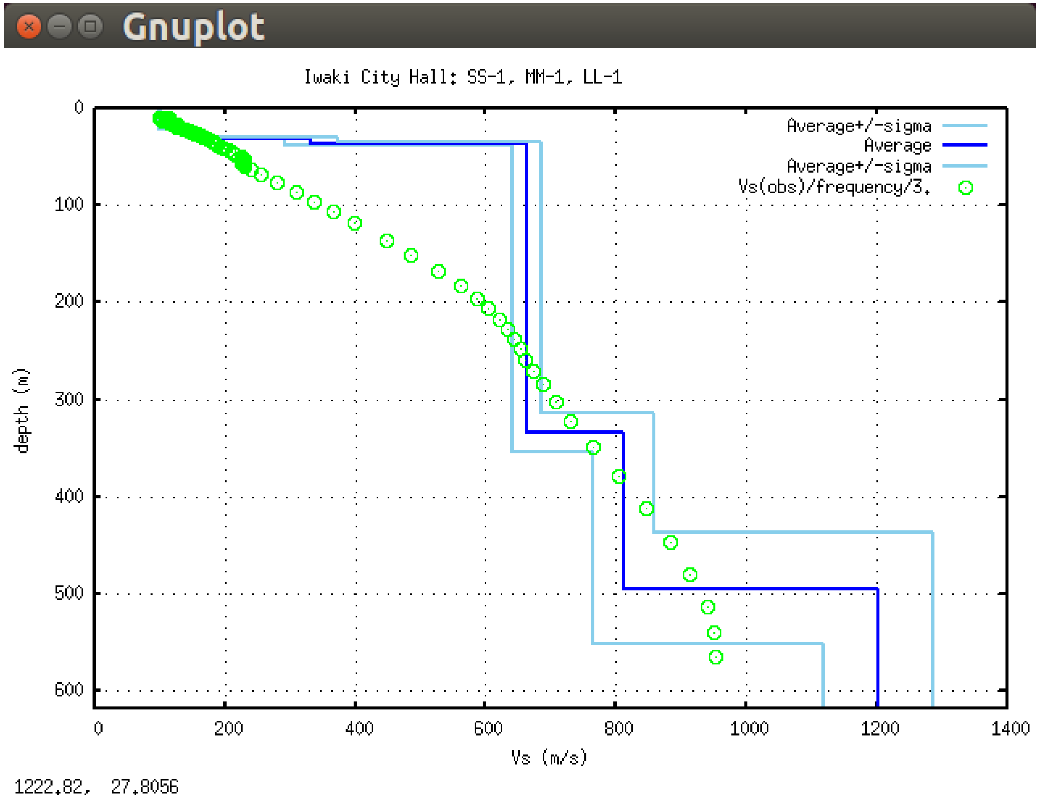
#	Frequency(Hz)	Observed Velocity	Calculated Velocity
0.562	0.952	0.966	0.165
0.586	0.950	0.946	0.158
0.610	0.941	0.923	0.153
0.635	0.914	0.895	0.140
0.659	0.883	0.867	0.130
0.684	0.847	0.837	0.117
0.708	0.804	0.809	0.107
0.732	0.766	0.782	0.098
0.757	0.731	0.756	0.090
0.781	0.709	0.734	0.084
0.806	0.687	0.713	0.079

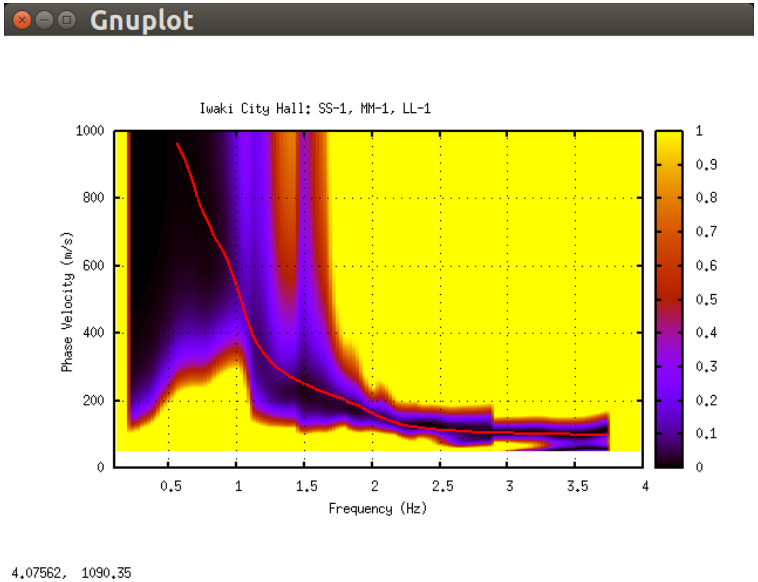
...

Graphical Output: *sh results.sh*

./spacwkf/prm/gnuplt_script/disp_cal.plt
./spacwkf/prm/gnuplt_script/cal_cf_panel.plt
./spacwkf/prm/gnuplt_script/vs_structure.plt
Hit return to continue
Hit return to continue



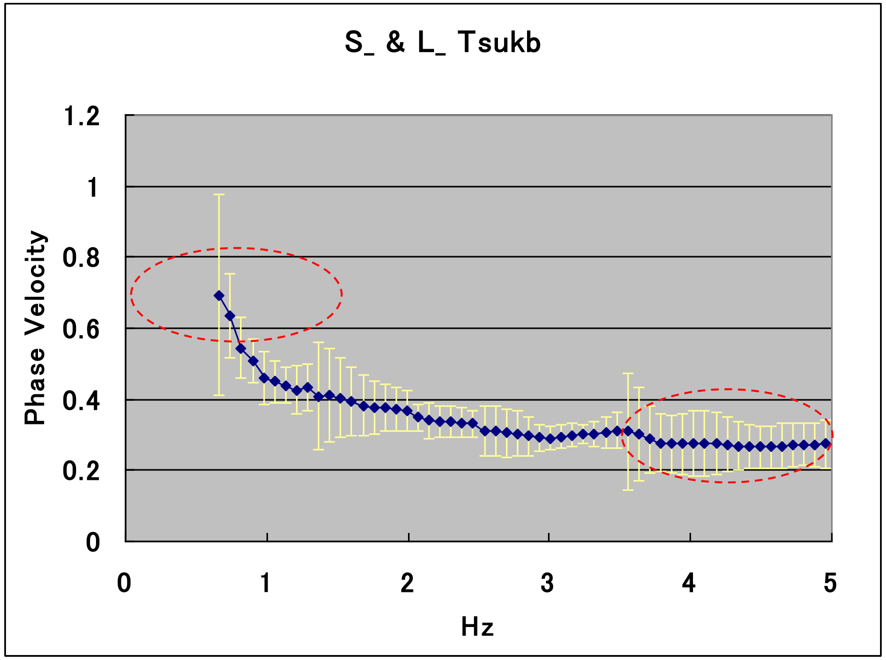




Set n_mod=0 in ‘vel_model_plt.prm’ to eliminate green circles.

These figures are automatically stored as PS files in ./spacwkf/data/results/fig_results

A way to find the optimum solution
- Explanation using another example -



In the highest frequency range, the phase velocity is about 0.3 (Km/sec).

At the lowest frequency 0.664 (Hz) the phase velocity estimated is 0.695 (Km/sec). Then, the corresponding wave length is about 1 (Km) and the expected explored depth very roughly estimated may be about 250 (m).

Examples of Search Range

str_range.dat (4 layer model for exploration from surface to the seismic bedrock)

```
Yamanaka (2001)                      :Model(a30)
    4                                :IL(I5),Layer Number
1.8 1.956 0.001 0.05  0.4  0.9  :density,Vp,hmin,hmax,vmin,vmax
2.0 2.400 0.001 0.30  0.7  1.3
2.3 2.955 0.010 0.30  1.2  1.8
2.5 4.842 998.0 999.0 2.6  3.6
```

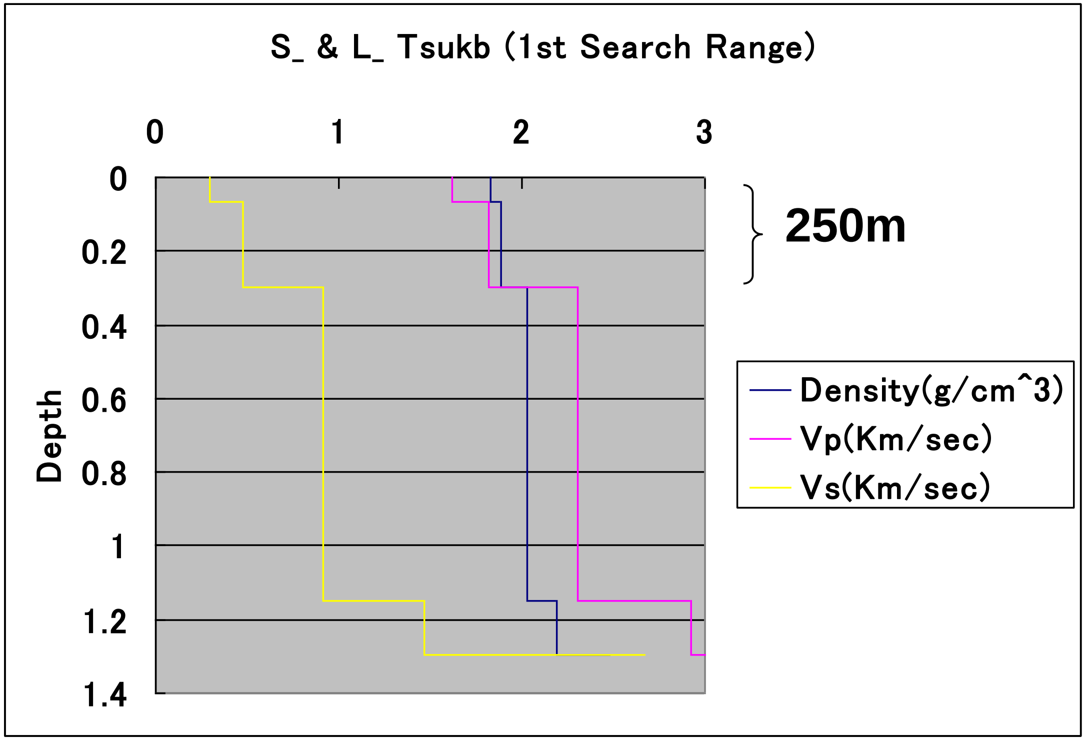
str_range.dat(4 layer model for exploration from surface to the engineering bedrock)

```
Engineering Bedrock                  :Model(a30)
    6                                :IL(I5),Layer Number
1.5 1.5  0.0   0.03  0.08 0.15  :density,Vp,hmin,hmax,vmin,vmax
1.5 1.5  0.001 0.03  0.10 0.15
1.5 1.5  0.001 0.03  0.08 0.15
1.6 1.5  0.001 0.03  0.15 0.25
1.7 1.6  0.001 0.03  0.25 0.35
1.8 1.8  998.0 999.0 0.35 0.8
```

1st Search Range

```
S_ & L_ Tsukb                          :Model(a30)
    5                                  :IL(I5),Layer Number
1.5   1.6   0.001 0.2   0.2  0.5 :density,Vp,hmin,hmax,vmin,vmax
1.8   1.956 0.01  1.0   0.4  0.9
2.0   2.4   0.1   1.0   0.7  1.3
2.3   2.955 0.1   1.0   1.2  1.8
2.5   3.2   998.0 999.0 2.6  3.6
```

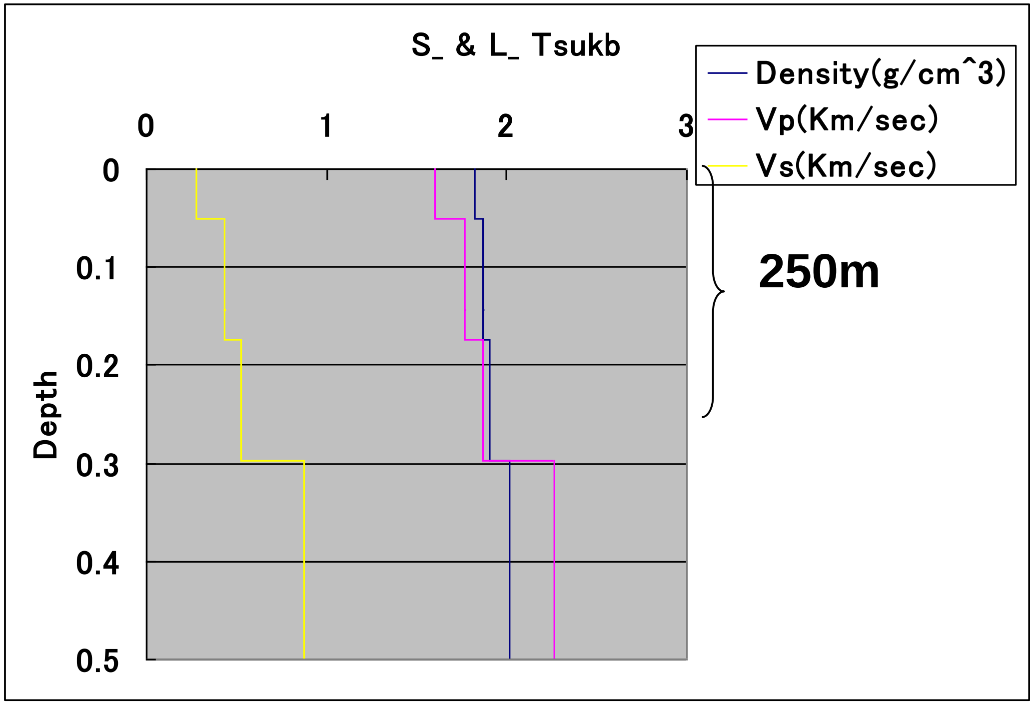
eps=0.01



2nd Search Range

```
S_ & L_ Tsukb                          :Model(a30)
    5                                  :IL(I5),Layer Number
1.5   1.6    0.001 0.2   0.2  0.4  :density,Vp,hmin,hmax,vmin,vmax
1.5   1.6    0.001 0.2   0.3  0.5
1.5   1.6    0.001 0.2   0.3  0.5
1.5   1.6    0.001 0.2   0.4  0.6
1.8   1.956  998.0 999.0 0.5  0.9
```

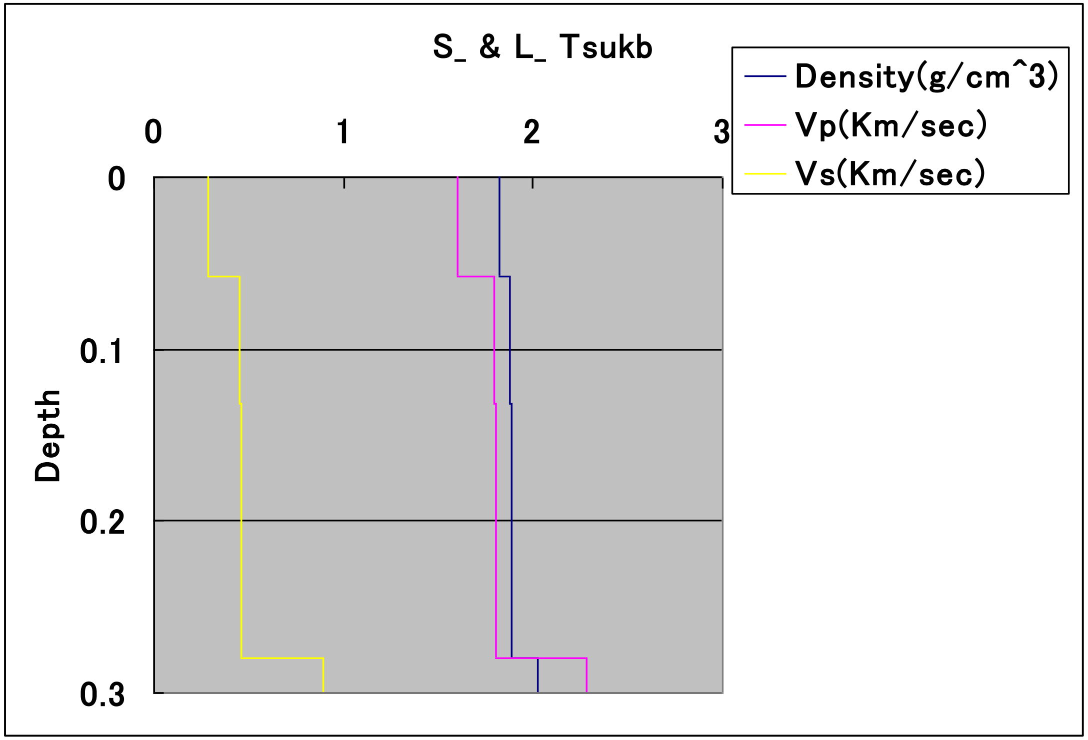
eps=0.0065



3rd Search Range

S_ & L_ Tsukb						:Model(a30)
4						:IL(I5),Layer Number
1.5	1.6	0.001	0.2	0.2	0.4	:density,Vp,hmin,hmax,vmin,vmax
1.5	1.6	0.001	0.2	0.3	0.5	
1.5	1.6	0.001	0.2	0.4	0.6	
1.8	1.956	998.0	999.0	0.5	0.9	

eps=0.0065



disp_sma1_2.prm

```
 1 1.  0.6 1.3 500   5   :idum,t0,a,c,ntemp,j0
 0.0065                    :eps0
    1    1                :n_roh,n_vp
    1    0    1           :ini_flg,ndsp_flg,n_err
    0    1                :kflg,jflg
    0    0                :n_vs,n_th
str_range.dat             :File name for the initial velocity model (a25).
vel_model.dat             :File name for the obseved dispersion relation (a25).
vel_cal.dat               :File name for the estimated velocity structure (a25)
disp_cal.dat              :File name for the calculated dispersion relation (a25)
```

c idum   :Random seed (integer)

c t_0 :Initial Temperature

c a,c   :Coefficients for $T=T_0*exp(-c*k**a)$, where k is iteration number

<The optimum schedule is given t_0=1.0, a=0.6, c=1.3 (Yokoi(2006)).>

c ntemp   :Maximum number of temparature change

c j0   :Number of iteration for each temperature

c threshold for conversion

c eps0   : averaged deviation

disp_sma1_2.prm (continuation)

```
c flags for roh and vp
c  n_vp     : 1=by Kitzunezaki et al(1990), vp=1.11*vs+1.29
c                 0=fixed to the initial values
c  n_roh    : 1=by Ludwig et al(1970), roh=1.2475+0.399*vp-0.026*vp**2
c                 0=fixed to the initial values
c flags for output to Display
c  ini_flg  : Initial Velocity Structure Model             1=yes
c  ndsp_flg : Observed Dispersion Relation                 1=yes
c  n_err    : Error at each iteration                      1=yes
c  kflg     : Missfit at each temp. change                 1=yes
c  jflg     : Missfit at each itration with the same temp. 1=yes
c  n_vs     : Vs value        (n_vs=layer number, 0=no output)
c  n_th     : Thickness value (n_th=layer number, 0=no output)
```

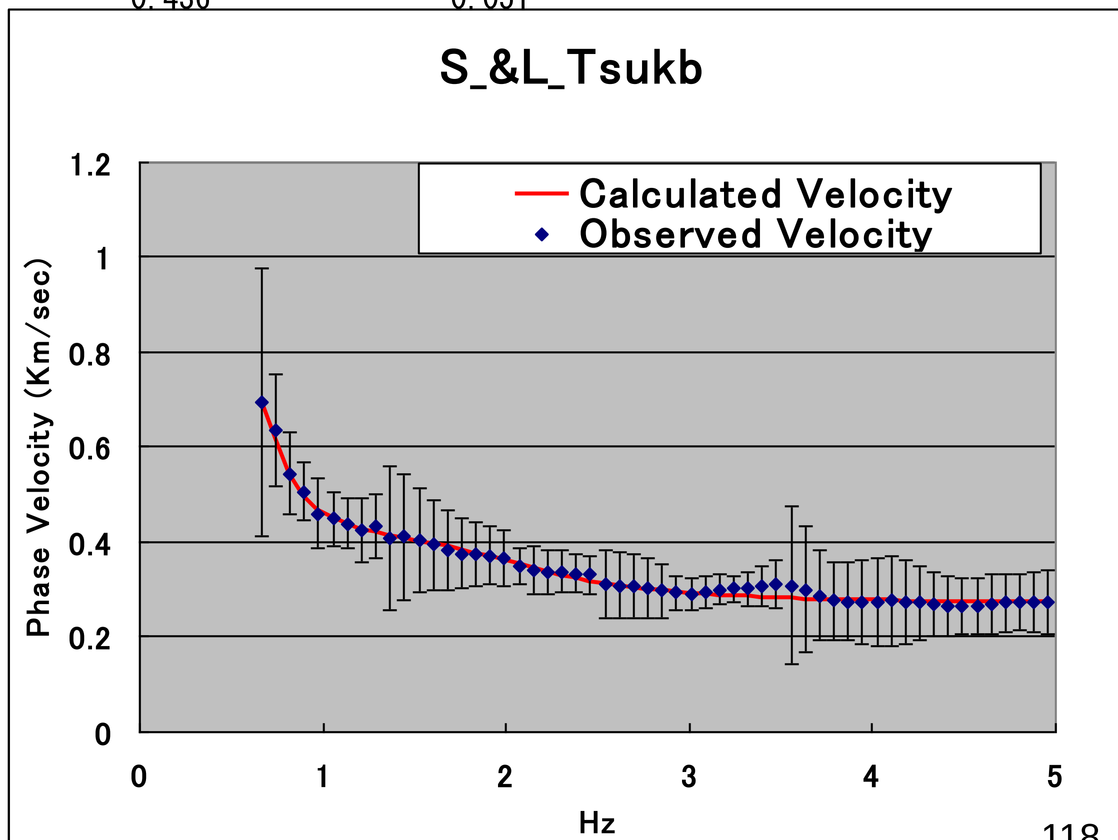
```
S_ & L_ Tsukb                           :Model(a30)
    4                                   :IL(I5),Layer Number
1.5   1.6    0.001 0.2   0.2  0.4  :density,Vp,hmin,hmax,vmin,vmax
1.5   1.6    0.001 0.2   0.3  0.5
1.5   1.6    0.001 0.2   0.4  0.6
1.8   1.956  998.0 999.0 0.5  0.9
```

	Thicknes(Km)	Density(g/cm^3)	Vp(Km/sec)	Vs(Km/sec)
1	0.057347	1.820632	1.604092	0.282966
2	0.074116	1.878893	1.791601	0.451893
3	0.147796	1.882880	1.804653	0.463651
4	999.000000	2.021464	2.277869	0.889972

Comparison of Cal. to Obs.

disp_cal.dat

Frequency(Hz)	Observed Velocity	Calculated Velocity	
0.664	0.694	0.698	0.281
0.742	0.636	0.618	0.118
0.820	0.545	0.543	0.085
0.898	0.507	0.496	0.061
0.977	0.460	0.467	0.074
1.055	0.449	0.449	0.058
1.133	0.440	0.436	0.051
1.211	0.426		
1.289	0.434		
1.367	0.408		
1.445	0.411		
1.523	0.405		
1.602	0.394		
1.680	0.382		
1.758	0.376		
1.836	0.375		
1.914	0.372		
1.992	0.367		
2.070	0.350		
2.148	0.341		
2.227	0.337		
2.305	0.338		
2.383	0.334		
2.461	0.331		
2.539	0.312		
2.617	0.309		
2.695	0.306		
2.773	0.303		
2.852	0.297		
2.930	0.293	0.294	0.037
3.008	0.291	0.292	0.033



Determined Velocity Structure

vel_cal.dat

	Thicknes(Km)	Density(g/cm^3)	Vp(Km/sec)	Vs(Km/sec)
1	0.057347	1.820632	1.604092	0.282966
2	0.074116	1.878893	1.791601	0.451893
3	0.147796	1.882880	1.804653	0.463651
4	999.000000	2.021464	2.277869	0.889972