Standard MCAP-CR Loudspeaker System Simulator
Multiple-Case Solver Program Document

December 30th, 2014
(revised manual dated August 10, 2018)

Shigeru Suzuki

0. Purpose of Distribution

This program is distributed for ones who are interested in bass-reflex loudspeaker technologies.

Used algorithm in this program is spring-mass based equation of motion model, so that it does not model continuum substances like loudspeaker membrane. It means that basic knowledge of physics is required to evaluate simulation results.

Ones who wish to evaluate solely quality or preference of sound without having interested in technology itself are not allowed to use this program.

1. Terms and Conditions to Use this Program

It is assumed that anyone who use this program agreed to following terms and conditions.

i. Users understand and agrees to limitations of mathematical models and algorithms used in this program.
ii. Users understand that there are unknown bugs in this program.
iii. Users do not claim for any loss incurred by using this program.
iv. Users do not use this program for commercial purpose.
v. Users do not redistribute this program or revised program code by themselves without written permission of original copyright holder.
vi. Users understand that it is the most important to keep consistency that program does not depend on particular platform. In order to this purpose, users do not convert this program code to any Microsoft Visual Studio codes.
vii. Users agrees that this program is not used for purpose to evaluate quality or preference of sounds.

2. New Features

This release has the following features:

i. A number of design cases are sequentially solved for frequency responses.
ii. Simulation condition input files are divided to driver parameters, analysis condition, and design conditions.
iii. Some of debugging feature is available (not much practical yet).
iv. Friction factors of air ducts are integrated to single parameter and made a function of duct diameter and length (determining parameter guideline is not yet provided).

3. Unnecessary functions still remaining

Following functions are not necessary for this multiple-case solver version, still remaining:

i. Single-frequency input signal option
ii. Random input signal option
iii. Sweep input signal option

4. Output Files

Most output files are still remaining, but they are not very helpful.

Important output files are the followings:

- case_out.csv
  This file includes frequency response for each case. First line is frequency as x-axis variable. Second and greater lines are frequency response (SPL[dB]).
- dft_case_results.csv
  This file contains frequency response in dB and in Pa, phase, real and imaginary part of Discrete Fourier Transform calculation.

- debuglog.txt
  Some debugging levels can be selected at program execution. If verbose debugging option is chosen, file size of debuglog.txt becomes huge and not recommended. This debug log is provided for developer (may be solely myself).

4. Functions of this Program

- This program computes displacements of masses, i.e. driver's membrane and masses of airs involved in ducts of standard MCAP-CR loudspeaker system.
- Users have option to compute Discrete Fourier Transform
- Input signal is assumed force rather than power, current, or voltage. Sinusoidal wave, linear sweep or random signal is prepared to use as signal input.
- Calculated results are stored in ASCII text files.

User needs to read the program code “multi-case_r01.cpp” if one need to know more details.

Note: Revision by Users

Users are allowed to revise this program code if one wishes under Terms and Conditions in Chapter 1.

3. Standard Use of this Program

Users have to compile this program code by themselves in order to get execution file.

This program is written in C language (and a little bit C++ grammar) , so that it can be compiled using right compilers.

This source code was checked using Linux GNU Compiler and MinGW.

Users who want to use with Mac OS need to set up gcc.

Microsoft C++ is not at all recommended to use with. It creates native code that is not compliant with standard C or C++ language.

Free BSD, UNIX or VMS may be used to compile this program, but not tested by developer.

Where to Get Compilers

GNU Compiler Collection http://gcc.gnu.org/
MinGW Projects http://www.mingw.org/

3.1 Preparation for Compiling

Linux (recommended)

GNU C Compiler (GCC) is generally pre-installed. Check using the following command (rpm system only).

```
$ rpm -qa|grep gcc⏎
```

Windows

Users need to install MinGW, for program code compilation. Much information is available through internet.

Binary Windows native executable file for x86 is also enclosed. Users need a dll file named “libgcc_s_dw2-1.dll”, this dll file is installed if MinGW is installed. If MinGW is not installed and a user want to use the executable, download the dll file from somewhere and copy to the directory where the executable is located.

3.2 Compiling Program Code

Linux

- Common throughout all distributions -

Copy decompressed source file to working directory.
Start up terminal (i.e. X Term) windows and move to working directory.
Enter gcc command to compile the code.

- Examples -

Vine Linux 6.0
$ g++ multi-case_r01.cpp⏎

"a.out" is given execution file name and users may give any name if operating system allows.

Windows
Copy decompressed source file to working directory.
Start up command prompt window and move to working directory.

MinGW
g++ multi-case_r01.cpp⏎
MinGW compiler generates "a.exe" file.

3.3 Running the Program (Linux/Windows)

3.3.1 Prepare Input Files

Create the following files in the working directory using text editor:

a) driver_parameters.txt
b) analysis_option.txt
c) cases.txt.

These files are required to run the program executable. Please note input files for this release are completely different from previous versions'.

a) Example of driver_parameters.txt

This parameter file defines driver's parameter values.
Notes:
Any letters or tabs or linefeed may not be placed prior to data value.
Linefeed code may be any if compiler recognizes. Notepad.exe may be used for Windows system.
Only line feed code may be inserted between data values in a line.
Only numbers in ASCII code may be used as data.
All the values are in double precision.

6.0	1st line	effective membrane mass m0 (in gram unit)
4.0	2nd line	effective membrane radius r0 (in cm unit)
80	3rd line	lowest resonant frequency f0 (in Hz unit)
0.5	4th line	mechanical Q : Qm (no dimension)

b) Example of analysis_option.txt

maximum resolution bit is 10 (greater values are assumed to be 10)
Cp/Cv : if not equal 1, it is assumed to be 1.4 (adiabatic)
friction constant f_c: $c_j = f_c \frac{l_j}{d_j^2}$ (under research for appropriate value)
All the values are in double precision, except resolution bit is integer.

7	1st line	resolution in bit (this case, resolution =T/2^7)
1.4	2nd line	ratio of specific heat Cp/Cv (1.4: adiabatic; 1.0 isothermal)
1	3rd line	amplitude of force signal in [N]
0	4th line	friction constant of ducts
0	5th line	vector sum option (0: all ducts; 1: ignore internal ducts)
25	6th line	sweep rate [Hz/s]
1	7th line	debug level (1: standard; 2: verbose; 3: default)

c) Example of cases.txt

From left to right with TAB separator:

case number (arbitrary integer)

number of **sub-**chambers (max 24; integer) Do not include main chamber.

each chamber volume (in [litre]; double precision) **Begin with main chamber, then continue to subchamber(s)**

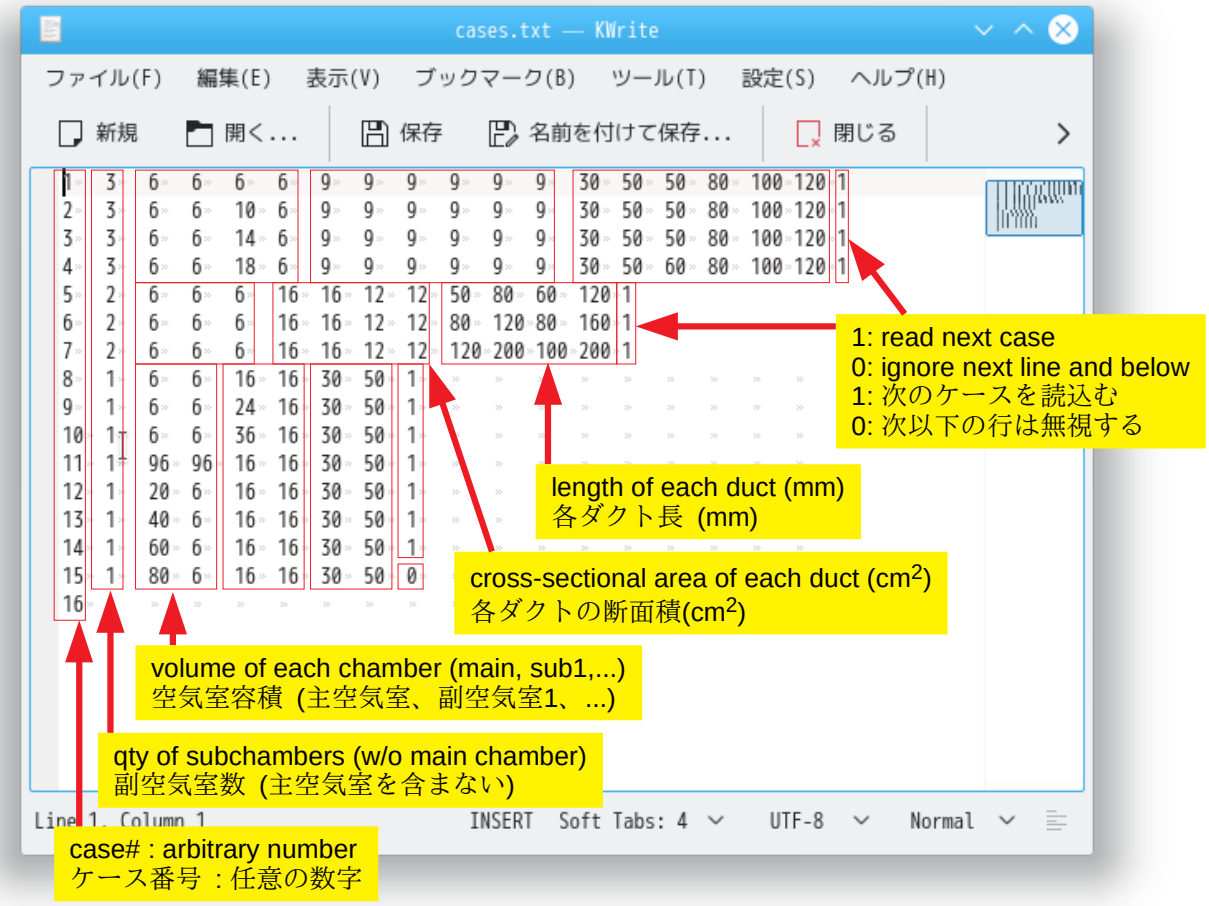
each duct area (in [cm^2]; double precision)

each duct length (in [mm]; double precision)

end mark (1: continue; 0: do not read below; integer)

Following cases show from top to bottom, double bass-reflex, standard MCAP-CR with two sub-chambers, and also standard MCAP-CR with two sub-chambers.

1	1	6.0	6.0	16.0	12.0	50.0	80.0	1					
2	2	6.0	6.0	6.0	16.0	16.0	12.0	12.0	50.0	80.0	60.0	120.0	1
3	2	6.0	6.0	6.0	16.0	16.0	12.0	12.0	80.0	20.0	80.0	160.0	0



3.3.2 Program Execution

Once source code was compiled, execution file may be used for other PC if platform is identical.

Move to working directory where binary executable file is located. Then type execution command to run the program.

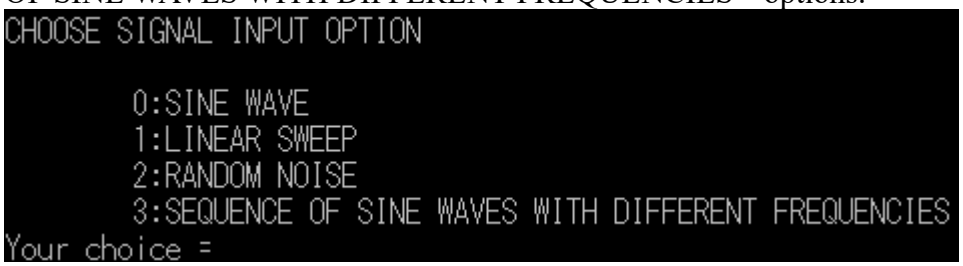
Linux (assuming execution file name is “a.out”)
$ ./a.out⏎

Windows (assuming execution file name is “a.exe”)
a⏎

Analysis Options

There are following options for analysis.

(1) Input Signal and Processing Options
Users may choose one of “SINE WAVE”, “LINEAR SWEEP”, “RANDOM” or “SEQUENCE OF SINE WAVES WITH DIFFERENT FREQUENCIES” options.



Refer to documents of previous versions for sine, sweep, or random options. These processes are the same as previous versions'. Author recommends option 3 only if a user choose this version.

1. Software set 10Hz and calculate DFT coefficient at 10Hz, then the value is recoded to “dft_case_results.csv” and “case_out.csv.”
2. Software set 10+1Hz and execute the same process as 1.
3. Software repeats adding 1Hz and execute same process up to 400Hz.

(2) Input Signal Option
Users may choose sinusoidal wave, linear sweep, or random input.
0: Sinusoidal wave in specified frequency
1: Linear sweep in specified frequency range
2: Random input signal
3: Sequence of sine waves with different frequencies (refer to above paragraph)

1: Linear Sweep
Specify final sweep frequency in Hz unit and sweep rate in Hz/s unit.
Maximum Frequency: 100 - 200 may be input. 200 is suggested value. Sweep rate shall be ristricted to 25[Hz/s] only.

2: Random
If this option is chosen, sampling frequency is set as 100 x number of divisions ($100 \cdot 2^{bit}$) in a cycle. No more input is required if this option is chosen.

Users should modify program code if other option is required.

3.3.3 Viewing Calculation Results

Output Files

Files given in Table-1 will be written in ASCII format.

Table-1 List of Output Files

File Name	Description	Note
case_out.csv	1st line : frequency 2nd line and below: case number, SPL in dB.	This is generated with option “3”.
dft_case_results.csv	Case number Frequency[Hz], SPL[dB], converted from dynamic pressure [Pa], phase[degree] at frequency, Real part of DFT, Imaginary part of DFT, ...	This is generated with option “3”.
dft_result.csv	Frequency[Hz], SPL[dB] converted from dynamic pressure [Pa], phase[degree] at frequency, Real part of DFT, Imaginary part of DFT	This is generated with option “0” thru “2”. Option.
normalized_x_vectors.csv	Time[s], Weighted sum of displacements[mm], displacement of membrane[mm], weighted displacement of air mass in a duct[mm],...	See below for definition of normalization. Not generated with option “3”.
normalized_v_vectors.csv	Time[s], Weighted sum of velocities[m/s], velocity of membrane[m/s], weighted velocity of air mass in a duct[m/s],...	See below for definition of normalization. Not generated with option “3”.
dftv.txt	1st line: number of datapoints 2nd line: time resolution 3rd line and below: normalized sum of velo1ocities	Used for Fourier Transform

The most hi-lighted output files are “case_out.csv” and “dft_case_results.csv” for this program revision.

Definition of Normalization

Following definitions are used for normalization.

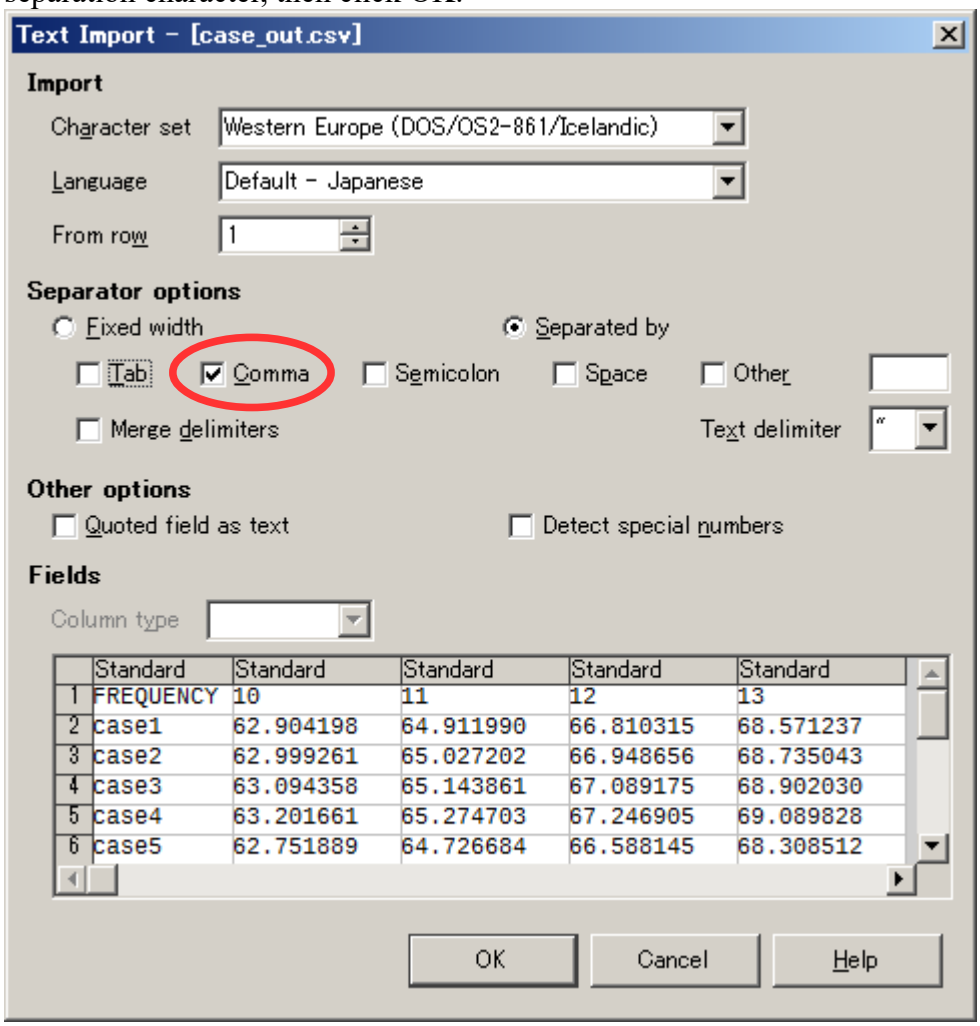
$x_j^* = r_j x_j = \frac{a_j}{a_0} x_j$ Displacement $v_j^* = r_j v_j = \frac{a_j}{a_0} v_j$ Velocity

Plotting Data

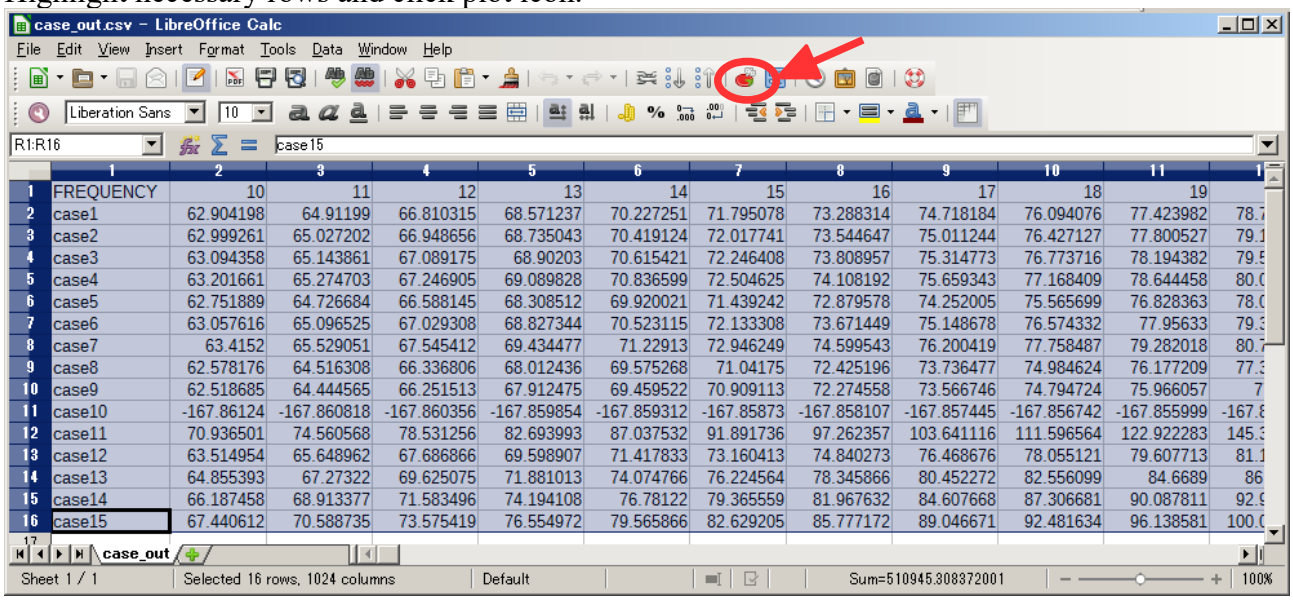
This program does not have plotting function. Users may use LibreOffice Calc or other application software to view data.

Example to Use LibreOffice Calc

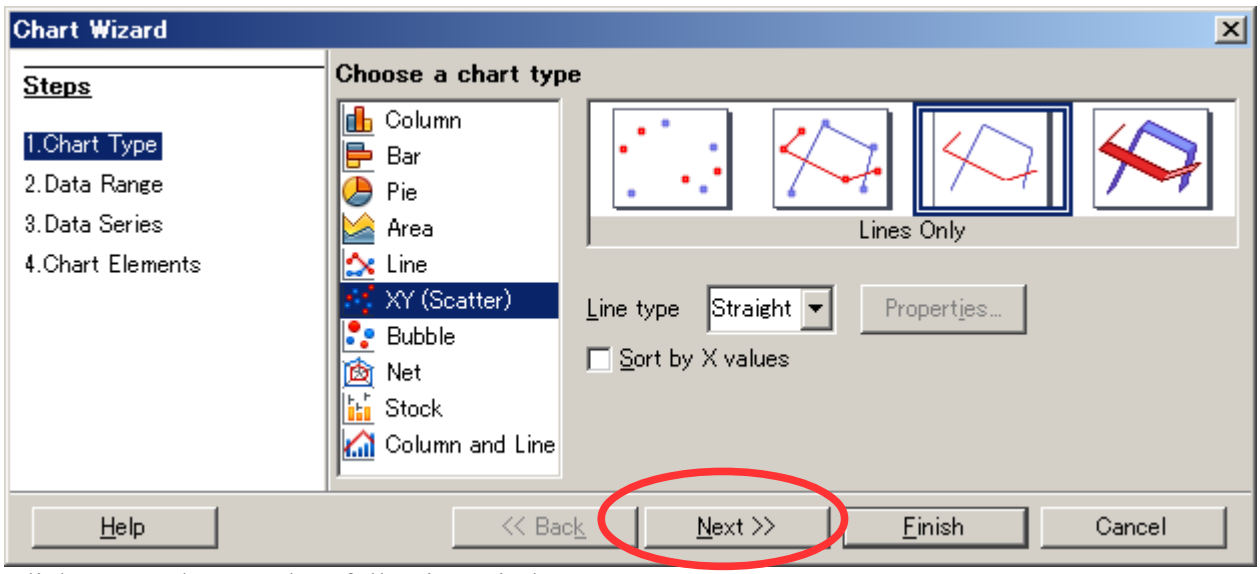
Open working directory using Windows Explorer or X Window Dolphin and double-click "case_out.csv". Calc starts up and display the following window, then select "Comma" as separation character, then click OK.



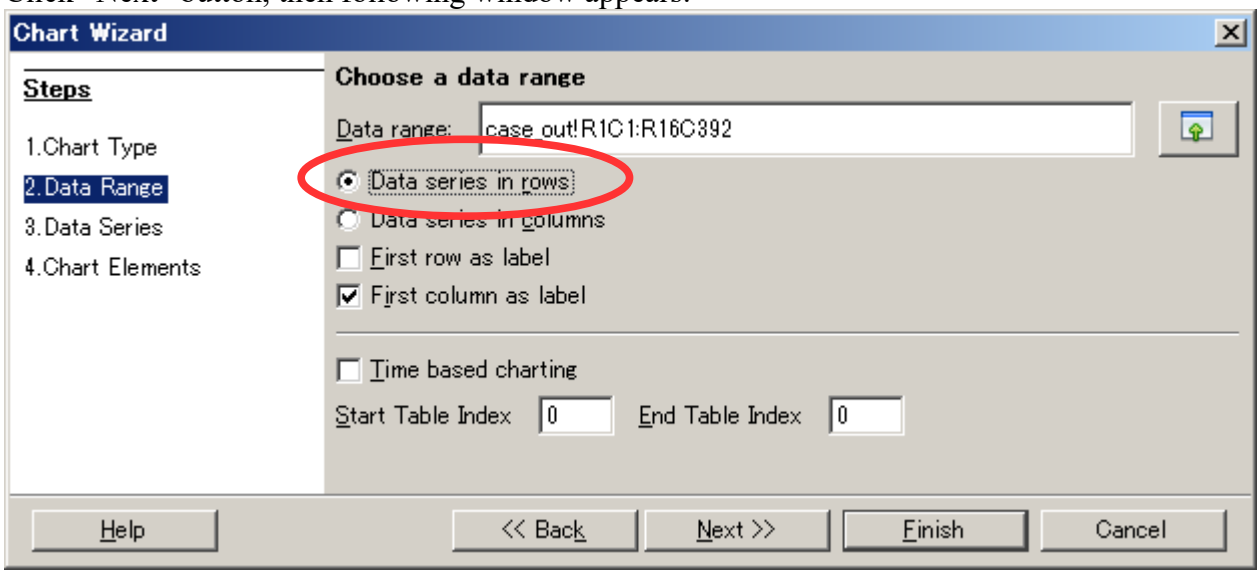
Highlight necessary rows and click plot icon.



Scatter option should be chosen in the following window.

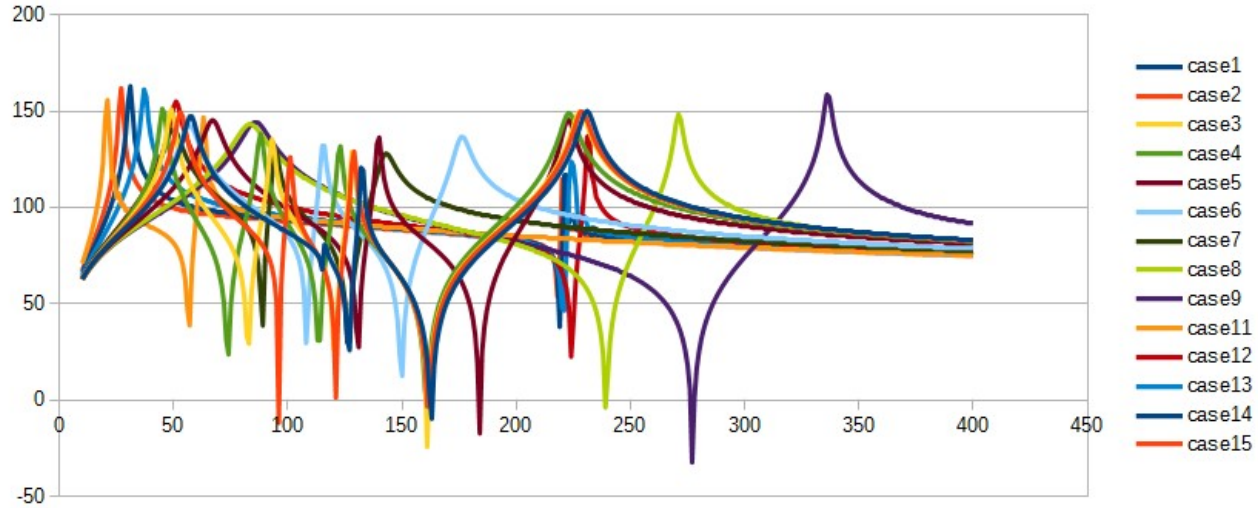


Click “Next” button, then following window appears:

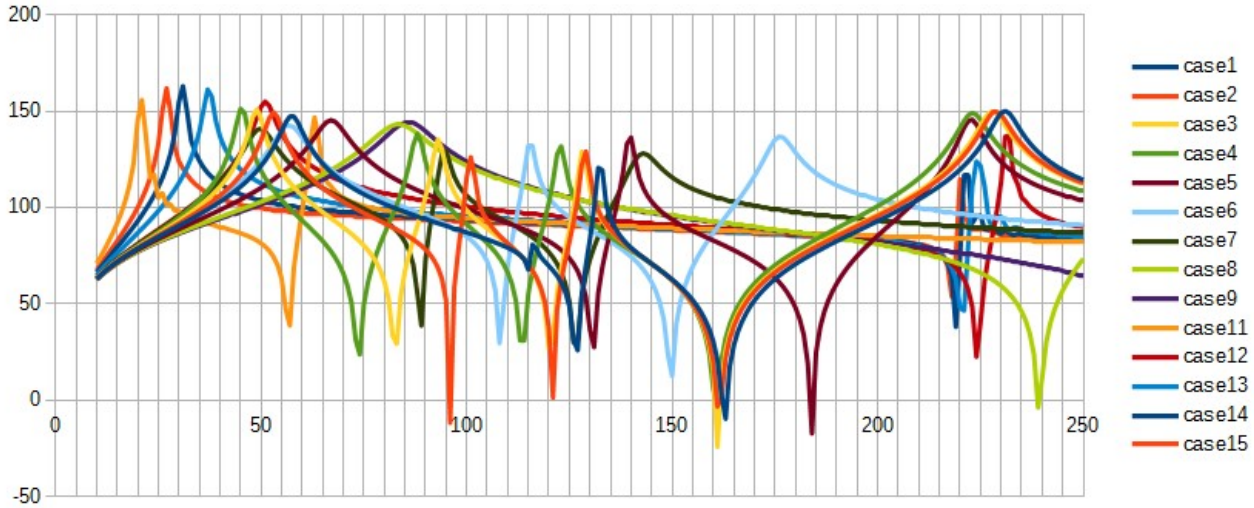


Click “Data series in rows”. This is the most important difference from previous versions'.
Then click “Finish button”.

Following plot then appears.



Adjust frequency range and insert vertical lines as you want to view.



Note that this is suitable to find characteristic frequencies, but may not be suitable for simulating frequency response. In order to view better frequency response curve, you need to determine the right friction constant.

This program uses traditional damping vibration model:

$$\frac{d^2x}{dt^2}+2\zeta\omega_n\frac{dx}{dt}+\omega_n^2x=\frac{f}{m} \quad .$$

Following model :

$$\frac{d^2x}{dt^2}+2\frac{c_f}{m}\left(\frac{dx}{dt}\right)^2+\omega_n^2x=\frac{f}{m}$$

may have better presentation of actual behavior.

In this case revision of friction model is required.

I would appreciate user's reports, if one modified the friction model. It will make it faster to complete the software.

Report any bugs to following email address, although it is not required.

mcapspeakers@gmail.com