Knowledge Base

Information

QA Article:	QA
Last Update Date:	03.
Rating:	
Views:	41

Creating WAV files from the results of an acoustic analysis

Portfolio / Domain: SIMULIA Abaqus Unified FEA / SIMULIA Abaqus Unified FEA
Product: n/a

QUESTION

I would like to create a ".wav" file so that I can "hear" the results of my acoustic analysis. Are there any utilities available?

ANSWER

(The following applies to Abaqus Versions 6.7 and higher.)

An Abaqus/CAE plug-in is available for this task and is attached to this Answer. The scripts are meant to be used after one of the following acoustics analyses: `*STEADY STATE DYNAMICS`, `*DYNAMIC` or `*DYNAMIC, EXPLICIT`. For `*STEADY STATE DYNAMICS`, the pressure (POR) results should be available as field data and in cases of `*DYNAMIC` and `*DYNAMIC, EXPLICIT` the pressure (POR) results should be available as history data or field data, in the output database (`.odb`) file. To request that this output data be written to the `.odb` file, include the following options in the steady state dynamics `*STEP` definition(s):

```
*OUTPUT, FIELD
*NODE OUTPUT, NSET=NodeSetName
  POR
```

or the dynamic `*STEP` definition(s)

```
*OUTPUT, HISTORY, FREQ=1
*NODE OUTPUT, NSET=NodeSetName
  POR
```

The script will extract the magnitude of the pressure at a given node as a function of time for the steady state dynamics field output data. With the user-supplied sampling rate, this function will be sampled at discrete points to create a table of pressures at these points. History data provides pressure points directly and they are also sampled with the user-supplied sampling rate.This data is then converted to an audio file in the WAV format. The higher the sampling rate used, the greater the resolution. Optionally, an XY Plot of the pressure curve can be saved.

The generated .wav files can be quite large, so the sampling rate must be chosen carefully. Since the results are harmonic, limiting the sampling rate to 2.5 times the highest frequency (in cycles/sec) in the model should be sufficient. The overall duration of the .wav file must also be specified, so limiting the overall duration to just slightly more than the reciprocal of the lowest frequency of interest (in cycles/sec) should be sufficient. In case of `*DYNAMIC` and `*DYNAMIC, EXPLICIT` procedures total length of the sound clip is determined by the total Step time. If user supplied frame rate is much smaller than the maximum frequency (i.e. 1/stable time increment) in the explicit analysis, neither the generated XY Plot of the pressure curve nor the .wav file may be accurate because of loss of data points.

If .wav files will be used to compare results of two separate analyses, care must be taken to properly account for the pressure scaling used within the script. By default, the pressure amplitudes at the given node are scaled such that the minimum and maximum values are bounded by -32768 and 32768 (16-bit resolution). The implication of this is that for a given volume setting on the playback software, the playback volume level of any two .wav files will be the same, independent of the actual (unscaled) pressure values. Appropriately changing the scaling factor will allow for an accurate aural comparison.

To do this, the value of `deltaY` used in the script must be scaled. As an example, consider two analyses. The first analysis will have a pressure range at the given node defined by

$$\Delta p_1 = p_1^{max} - p_1^{min}$$

a similar quantity can be defined for the second analysis. The range of pressure at the given node can be determined by examination of the XY plot generated the first time the script is run. Referring to the smaller pressure range as Δp_{low} and the larger pressure range as Δp_{high}, `deltaY` is properly scaled as

$$\mathtt{deltaY}_{Scaled} = \mathtt{deltaY}*(\Delta p_{low}/\Delta p_{high})$$

Executing the script a second time, with the scaled value of `deltaY` and the `.odb` file of the analysis with the lower pressure range, will produce a .wav file with the appropriate relative volume scaling.

The attached simple one-element deck `create_audio_file.inp` illustrates the script functionality. Node 1 is driven at a specific frequency in each of four `*STEADY STATE DYNAMICS` steps. Run this analysis first. Once the analysis is complete, follow the Version specific instructions outlined below. The attached figure `create_audio_file.png` shows the pressure at node 2 as a function of time for the first 0.1 sec. It can clearly be seen to be a combination of several frequencies. Musically it is the time signature of the Cm7 chord. The attached file `create_audio_file.wav` contains a 1 second segment of this time history.

Limitations

- The script provides only mono output and is limited to one byte per sample.
- The script is designed for use with acoustic analyses conducted with the `*STEADY STATE DYNAMICS`, `*DYNAMIC` and `*DYNAMIC, EXPLICIT` procedures only.
- The script cannot be run with .odb files generated from models in parts and assembly format.

Installation

To install the plug-in, download and save the attached archive to one of the following directories:

- *abaqus_dir*\abaqus_plugins where *abaqus_dir* is the Abaqus parent directory
- *home_dir*\abaqus_plugins where *home_dir* is your home directory
- *current_dir*\abaqus_plugins where *current_dir* is the current directory

Note that if the `abaqus_plugins` directory does not exist in the desired path, it must be created. The *plugin_dir* directory can also be used, where *plugin_dir* is a directory specified in the `abaqus_v6.env` file by the environment

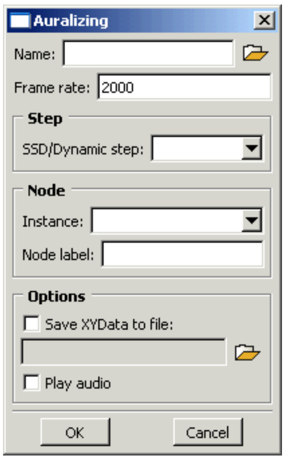
MY FAVORITE CONTENT

variable **plugin_central_dir**. You can store plug-ins in a central location that can be accessed by all users at your site if the directory to which **plugin_central_dir** refers is mounted on a file system that all users can access. For example,

```
plugin_central_dir = r'\\fileServer\sharedDirectory'
```

On Windows platforms, right click on the archive file and select **WinZip** → **Extract to here**. On Linux platforms, type **unzip CreateAudioFile.zip** at the command prompt. Note that the plug-in will not function properly if this procedure is not followed.

The next time Abaqus/CAE is started, a menu item named **Auralizing...**Plug-ins pull down menu from the **Visualization** module. Select **Plug-ins** → **NVH** → **Acoustics Toolset** → **Auralizing...** to invoke the plug-in dialog box:



Revision History

14 Apr 2006	Documentation link in plug-in corrected. Updated version included in `CreateAudioFilePlug-In.zip`. Version increased to 1.1.
10 Oct 2007	Plug-in updated to include *DYNAMIC, EXPLICIT support. Release increased to 1.3-1.
17 Jun 2009	Fix *DYNAMIC, EXPLICIT bugs related to time calculation and other issues. Release increased to 1.3-2.
14 Aug 2009	Added support for *DYNAMIC and fixed some bugs related *DYNAMIC, EXPLICIT. Release increased to 1.3-3.
09 Nov 2009	Added an interpolation function to calculate POR from the history data and support for field output with *DYNAMIC and *DYNAMIC, EXPLICIT. Release increased to 1.3-4.
12 Jul 2010	Fix max values used in wave. Release increased to 1.4-1.
25 Jan 2011	As check for Numeric import for different releases. Add check if abs(pmin) > pmax. Release 1.4-2.

Disclaimer

KEYWORDS **sound, wav, wave, audio, plugin, plug-in, plug in, python, scripting, kernel, script, custom, custom**

ATTACHMENT create_audio_file.inp answer_1106_fig1_new.png CreateAudioFile.zip create_audio_file.wav

SUBSCRIBE TO CHANGES ☐

RATING **On a scale of 1-5, how would you rate the technical content of the article?**

Please rate this article...

LET US KNOW IF THIS ARTICLE NEEDS TO BE ENHANCED

UNCLEAR MISSING INFO DUPLICATE OUT OF DATE ERROR DETECTED

See Comments (1)

MY FAVORITE CONTENT