Knowledge Base

Creating Sound Transmission Loss Output from Abaqus Acoustics Analyses

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

Information

QA Article: QA(
Last Update Date: 29.(
Rating:
Views: 583

MY FAVORITE CONTENT

QUESTION

I have run an acoustics analysis in Abaqus/Standard and I would like to create sound transmission loss (STL) output. How can I do this?

ANSWER

Two Abaqus/CAE plug-in applications for this purpose are attached below.

The first application, named **Muffler STL,** uses POR output. The typical use case is simulation of an expansion muffler in which a plane wave propagates from one end to the other.

The second application, named **STL**, uses INTEN output. The typical use case is quantifying STL of an automotive barrier. In a general test setting, the barrier material is placed between two reverberant rooms. The acoustic power at any point inside a reverberant room is uniform.

Each application is independent of one another and is to be used for different purposes. Instructions are given below for the **Muffler STL** application, followed by those for the general **STL** plug-in.

(The following applies to Abaqus Version 6.8 and later)

Muffler STL Installation

To install the plug-in, save the attached archive file 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 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,

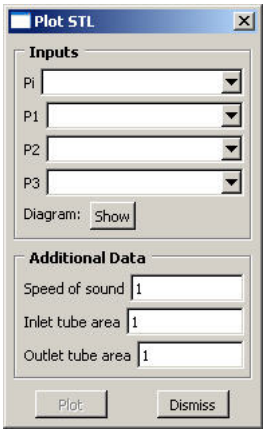
plugin_central_dir = r'\\fileServer\sharedDirectory'

On Windows platforms, right click on the archive files and select **WinZip → Extract to here**. On Linux platforms, type **unzip MufflerSTL_plug-in.zip** at the command prompt. A folder named abq_MufflerStl and a file named mufflerStl_plugin.py will be extracted.

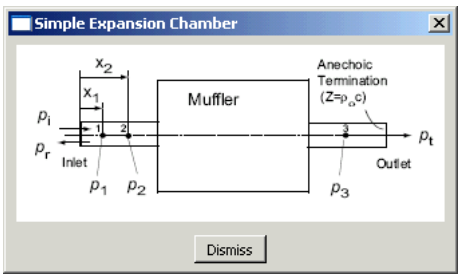
Note that the plug-in will not function properly if this procedure is not followed.

Muffler STL Usage

In Abaqus/Viewer or the Visualization module of Abaqus/CAE select **Plug-ins → NVH → Acoustics Toolset → Plot STL using POR...** to receive the following dialog:

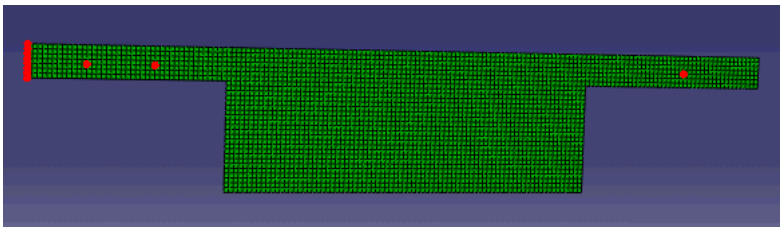


The "Show" button displays a simple expansion chamber diagram that describes where points 1, 2 and 3 are located, which correspond to P1, P2 and P3.



The **Muffler STL** utility can only be used on results generated with the steady state dynamics procedure on a 2D simple muffler. The job must contain only a single step that runs a direct steady state dynamics analysis. Execute the steady state dynamics analysis and request acoustic pressure (Abaqus output: POR) as the field output. Pre-defined node sets

for points 1, 2 and 3 (one node per node set) and also for P_i (nodes at the inlet) must be available in the database. The nodes at the inlet are used to measure the distance of points 1 and 2. The node sets can be either created in the input file or in Abaqus/Viewer interactively by using the plug-in available in Simulia Answer Abaqus/CAE plug-in for creating node and element sets in the Visualization module. A sample muffler is shown below with the highlighted node sets in the viewport. It is assumed that a one dimensional plane wave propagates through the inlet side of the muffler.



Notes

The plug-in uses the three-point method to calculate STL, and it is defined as,

$$STL = 20\log_{10}\left\{\frac{|p_i|}{|p_3|}\right\} + 10\log_{10}\left(\frac{S_i}{S_o}\right)$$

where S_i, and S_o are the inlet and outlet tube areas respectively.

In the three-point method, STL is estimated from the POR measured at three points inside the simple muffler. Among the three points, two of them (1 and 2) are located on the inlet side and point 3 is located on the outlet of a pipe. The POR at points 1 and 2 are used to extract the incident pressure (P_i). Since the outlet has anechoic termination, it is assumed that POR at the point 3 is the same as the transmitted wave pressure (P_t) at pipe outlet, i.e., $P_3=P_t$. The POR at points 1 and 2 in the inlet pipe are resultant of both incoming (P_i) and reflected (P_r) waves and are given by,

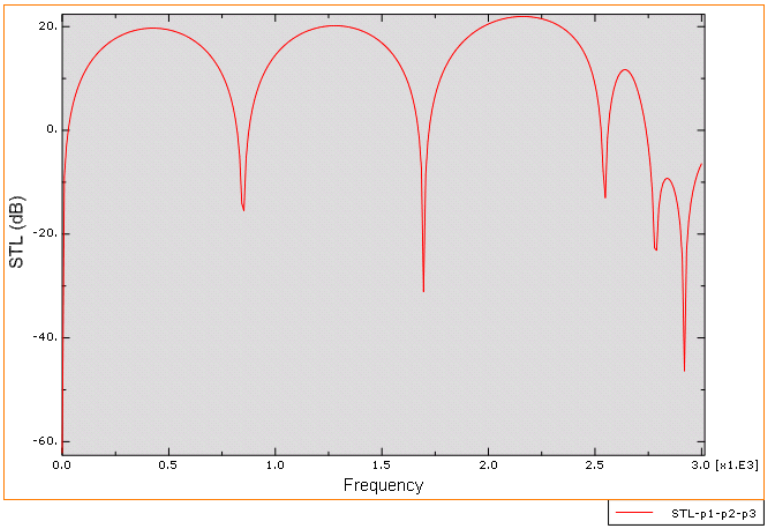
$$p_1 = p_i\, e^{ikx_1} + p_r\, e^{-ikx_1}$$

$$p_2 = p_i\, e^{ikx_2} + p_r\, e^{-ikx_2}$$

Where P_1 and P_2 are POR values, x_1 and x_2 show locations of points 1 and 2 respectively (as shown in Simple Expansion Chamber image), i=√-1. , k=w/c where w is angular frequency and c is speed of sound. Solving the above two equations we can obtain P_i,

$$p_i = -\frac{1}{2i\sin k(x_2 - x_1)}\left[p_1 e^{-ikx_2} - p_2 e^{-ikx_1}\right]$$

A sample STL plot output using this plug-in is shown below.



Where sink(x2-x1)≠0 or k(x2-x1)≠nπ, n=0,1,2,....

(The following applies to Abaqus Version 6.5 and later)

STL Installation

To install the plug-in, save the attached archive file 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 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'

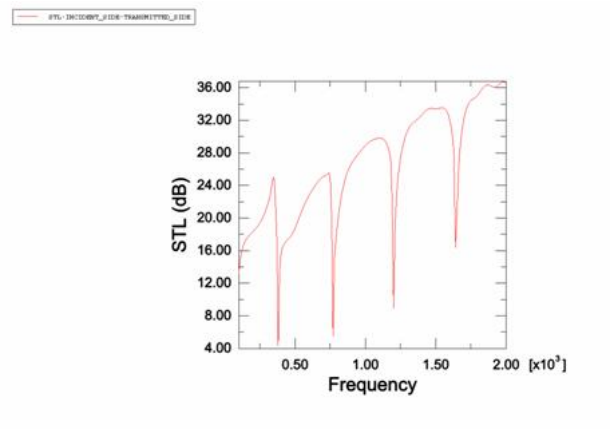
On Windows platforms, right click on the archive files and select **WinZip → Extract to here**. On Linux platforms, type **unzip STL plug-in.zip** at the command prompt. A folder named abq_sTL and a file named stl_plugin.py will be extracted.

Note that the plug-in will not function properly if this procedure is not followed.

STL Usage

The plug-in can only be used on results generated with the steady state dynamics procedure. Execute the steady state dynamics analysis and request the INTEN field element output (see attached sample model OneSeal.inp for details). Choose the appropriate element sets from the **Incident side element set** and **Transmitted side element set** drop-down menus .

A history output for STL is calculated and displayed in the current session.



Notes

"Sound transmission loss" (STL) refers to the reduction in amplitude of sound waves passing through a barrier. Barriers are typically engineered to suppress sound transmission across a particular frequency band; for example, an automotive panel may be designed to have a high STL at those frequencies associated with an engine or other sound-producing components.

There are two main mechanisms by which barriers reduce the amplitude of transmitted waves, resulting in higher STL: reflection and absorption. Reflection of airborne sound will be high for very massive and/or very stiff barriers, because the barrier is largely immobile in response to the incident acoustic sound pressure. The air on the transmission side, excited by the small motion of the barrier, will respond only at low levels, so that STL is high. The absorption mechanism is different: foam, viscous, or otherwise dissipative structures reduce the amplitudes of structural waves through their thickness. Nevertheless, the STL design goal for the barrier is the reduction of the amplitude of the normal velocity, acceleration, or intensity at the receiving side.

The STL is defined as

$$STL = 10 * \log_{10} \frac{\sqrt{I_i \cdot I_i^*}}{\sqrt{I_t \cdot I_t^*}}$$

where I_i (I_t) is the set of INTEN output calculated at the centroids of all the elements in the incident (transmitted) side element set and I_i^* (I_t^*) is the complex conjugate of I_i (I_t).

STL to far-field boundaries

The STL plugin as seen above helps in obtaining sound transmission loss within domain using the output INTEN. The script `acousticVisualization.py` from the Abaqus documentation makes available the corresponding output INTEN_FAR for a far-field (from infinite elements). Using INTEN_FAR output, the script `STL_farfield.py` helps in obtaining sound transmission loss to far-field boundaries. So essentially it applies the approach of the STL plugin to the output obtained from the `acousticVisualization.py` script. The attached archive `STL_farfield.zip` contains the script and example files along with a guidelines document.

Disclaimer

Muffler STL Revision History

9 Jul 09	Release of Version 1.1-1.

STL Revision History

01 Jun 07	Release of Version 1.1-1.
20 Jul 08	Release of Version 1.1-2. Changed register query and removed condition for combo box.
19 Aug 11	Release of Version 1.1-3. Modified curves to plot function.
27 Jul 18	Release of Version 1.1-4. Removed uti.getVersion() check.
29 Jul 20	Release STL plugin for far-field boundaries. Uploaded STL_farfield.zip

KEYWORDS **sound, transmission, output, acoustic, stl, loss, plug, plug-in, frequency, vibration, 3439**

ATTACHMENT
answer_3439_fig18.jpg | answer_3439_fig11.jpg | Answer_3439_Fig3.gif | answer_3439_fig13.jpg | answer_3439_fig14.jpg | answer_3439_fig16.jpg | MufflerSTL_plug-in.zip | Answer_3439_Fig2.jpg | answer_3439_fig12.jpg | OneSeal.inp | answer_3439_fig15.jpg | answer_3439_fig17.jpg | STL_plug-in.zip | STL_farfield.zip

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

MY FAVORITE CONTENT