

Knowledge Base

Information



Abaqus/CAE plug-in to compute the discrete Fourier transform of time history data

Portfolio / Domain: SIMULIA Abaqus Unified FEA / SIMULIA Abaqus Unified FEA
Product: SIMULIA Abaqus/CAE

QA Article: QA0000008451e
Applicable Level: 6.7
Last Update Date: 29/11/2017
Rating:
Views: 110

QUESTION

I have a set of time history output from a linear or nonlinear transient analysis. How can I convert the data from the time domain to frequency domain?

ANSWER

(The following applies to Version 6.7 and higher)

An Abaqus/CAE plug-in for this purpose is attached below. The plug-in allows you to plot the discrete Fourier transform of a set of time history data. The transformed data is available as an XY Data object.

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,

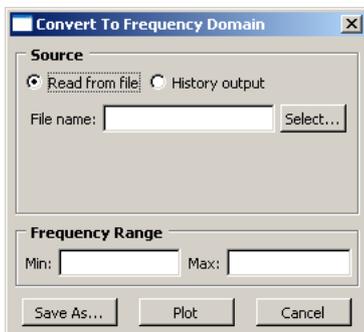
`plugin_central_dir = r'\\fileServer\sharedDirectory'`

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

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

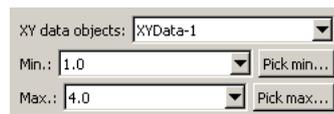
Usage

In Abaqus/Viewer or the Visualization module of Abaqus/CAE select **Plug-ins** → **NVH** → **Convert to Frequency Domain...** to receive the following dialog:



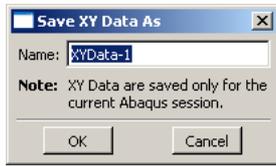
1. If the time history data is available as an ASCII file, select **Read from file**, then **Select...** to navigate to the desired file. The file must have the *.rpt* extension, and the data must be in two-column, space delimited format, with values of time in column one. The plug-in will skip any text or unreadable strings.

Or, if the time history data is available as an XY Data object, select **History output**, then select the desired XY Data from the drop-down list. Upon selection of an XY Data, the minimum and maximum time values of the selected XY Data will be shown. Manually select the minimum and/or maximum time values if desired, as shown in the following dialog:



You may also select the **Pick min...** or **Pick max...** buttons to select the minimum or maximum time value from the viewport. In order to this, the XY Data object must be plotted in the current viewport. Select from the viewport the time (x-axis) desired. At the present time, this utility does not allow you to select a specific point in the plotted curve. Therefore, you must select an arbitrary time value and the utility will find the closest minimum or maximum time value from the shown XY Data. The corresponding drop-down menu will be automatically updated.

2. The **Frequency Range** minimum and maximum inputs can be used to filter the output values. If the expected output frequency range is known before-hand and would like to be the range shown, enter the values for minimum and maximum, otherwise leave these inputs blank.
3. Select **Plot** to automatically plot the transformed data in the current viewport. Note that a temporary XY Data object is created to store the transformed data. Select **Save As...** to save the transformed data as a new XY Data object. The following dialog appears to give a new name for the XY Data object:



After selecting OK, the XY Data object will be listed in the XY Data Manager.

Calculations

For a given N-pair time history data with constant time step:

$$(t_1, p_1), (t_2, p_2), \dots, (t_r, p_r), \dots, (t_N, p_N)$$

the corresponding frequency-domain data

$$(f_1, P_1), (f_2, P_2), \dots, (f_s, P_s), \dots, (f_N, P_N)$$

is obtained using the discrete Fourier transform (DFT) as follows:

$$f_1 = 0, \quad P_1 = \frac{1}{N} \sum_{r=1}^N p_r$$

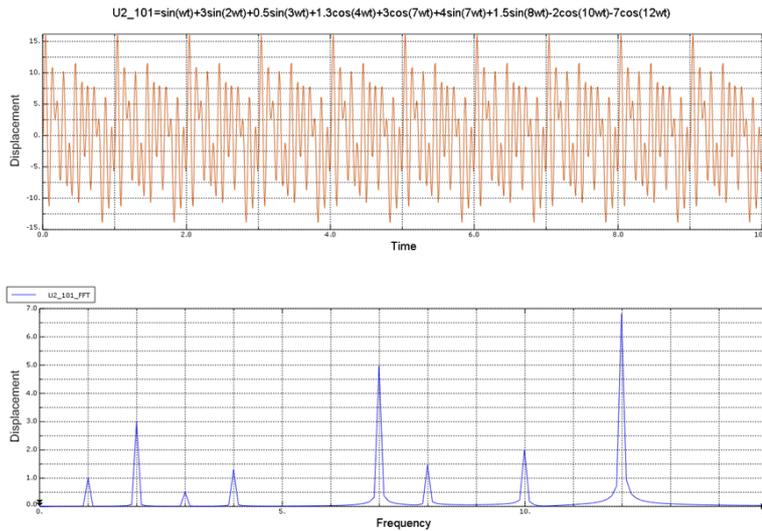
$$f_s = \frac{s-1}{N \Delta t}, \quad P_s = \frac{2}{N} \left| \sum_{r=1}^N p_r e^{2\pi i(s-1)(r-1)/N} \right|$$

where

$$\Delta t = t_2 - t_1$$

is the time step and s=2, 3, 4... Note that a fixed time increment will be used, taking the first two time values to calculate the increment. We recommend using Abaqus/Viewer **Interpolation** functionality from the **Operate on XY data** menu in order to obtain data adequate for this utility. Please refer to section 29.4.14 **Interpolating an X-Y data object** of the Abaqus/CAE User's Manual for more information.

For example, a periodic function and its DFT are shown below:



It is seen in the FFT that there is a peak at each of the frequencies represented in the expression.

Notes

1. This utility uses the FFT.py module Discrete Fourier Transforms module available within Abaqus. The underlying code for these functions is an f2c translated and modified version of the FFTPACK routines. See [Numeric FFT](#) for details.
2. The XY data to be transformed is not required to contain a certain number of points.

Disclaimer

The attachments to this article are subject to certain usage conditions. Please [click here](#) for details.

Revision History

| | |
|-----------|--|
| 16 Dec 08 | Release of Version 1.1-1 |
| 06 Apr 09 | Release of Version 1.2-1. Using Numeric FFT implementation. |
| 24 Apr 09 | Release of Version 1.3-1. Allow for frequency range user input. Remove slider and give drop-down menus for time range. Also allow to pick from viewport for min/max time values. |
| 05 May 09 | Release of Version 1.3-2. Check y-axis type when using XY data. |
| 08 Dec 09 | Release of Version 1.3-3. Disable viewports picks when xy data is not active in viewport, allow frequency values to be removed once specified, and allow any ASCII file to be read in. |
| 07 Feb 11 | Release of Version 1.3-4. Support numpy module. |

ATTACHMENT



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 \(4\)](#)