Abaqus2PragTic
User Documentation

authors: Martin Nesládek
Jan Papuga
Martin Hronek

address: Faculty of Mechanical Engineering, CTU in Prague
Department of Mechanics, Biomechanics and Mechatronics
Technická 4
166 07 Praha 6
Czech Republic

Contents

1 Introduction

The following text describes features and functions of the Abaqus2PragTic I/O interface denoted for data transfer between Abaqus FE solver and PragTic[1] fatigue postprocessor. The document gives information about installation process, concept of the data exchange between the programs and discusses limitations of the current approach.

2 Licensing

The provided source codes can be used, modified and distributed for any purpose and free of charge. The authors are not liable for any damage or data loss caused by use of the program.

3 Installation and configuration

Functionality of the interface has been tested on linux platforms (Fedora distribution) with Abaqus 6.8 to 6.11 versions until now. But since the code is written in Python language, it is supposed to be a cross platform application. The only prerequisite is Abaqus/CAE installed on the computer which is intended for storing the FE model data in *.odb* databases. The steps required for successful Abaqus2PragTic installation are as follows:

1. Create *abaqus_plugins* folder in your home directory.

2. Extract all files from *Abaqus2PragTic.zip* and copy the GUI files *abaPr.py*, *abaPr_plugin.py* and *abaPr_conf.py* into *abaqus_plugins*. The core files *Abaqus_Pragtic.py* and *Pragtic_Abaqus.py* can be stored elsewhere.

3. Open the *abaPr_conf.py* and provide changes to the following variables:

   - *FPU_HOSTS* – a list of possible computer hostname strings, which are intended for running the application. If Abaqus2PragTic will run only on a single computer, provide one item to the list. If more computers are planned for running the application you can enter all these names in the list at once and then just make a copy of it to other computers without changing it.
   - *FPU_ABAPR_PATH* – a string specifying a directory path to *Abaqus_Pragtic.py* and *Pragtic_Abaqus.py* files. These files can be stored in other directory than *abaqus_plugins*.
   - *FPU_ABAQUS_PATH* – a location of Abaqus installation, e.g. *"/usr/abaqus/6.11-3"*.

[1] www.pragtic.com

4 Concept of data transfer between Abaqus and PragTic

Data flow between Abaqus and PragTic is summarised by chart in Fig. 1. In the beginning of the export/import procedure we assume existing *.odb* database containing stress analysis results. User then defines entities for export in Abaqus2PragTic interface. A copy of user input from widgets is stored in *.fpu* file. The core script then creates *.pti* file based on information in *.fpu* file. *.pti* file is a detailed list of the selected model topology, geometry sets and results.

On the PragTic level *.pti* file is read by import procedure and data are translated to standard PragTic database. Based on the data, a fatigue analysis can then be performed and if needed, the obtained results can be exported to *.pto* file for subsequent visualization in Abaqus/Viewer. The *.pto* file data are imported to *.odb* database by another core script of Abaqus2PragTic interface.

NOTE: Procedures for data manipulation on the PragTic level (section 5.3 and 5.4) are not part of this distribution. These are components of PragTic binaries and can be downloaded only as a complete PragTic package from www.pragtic.com.

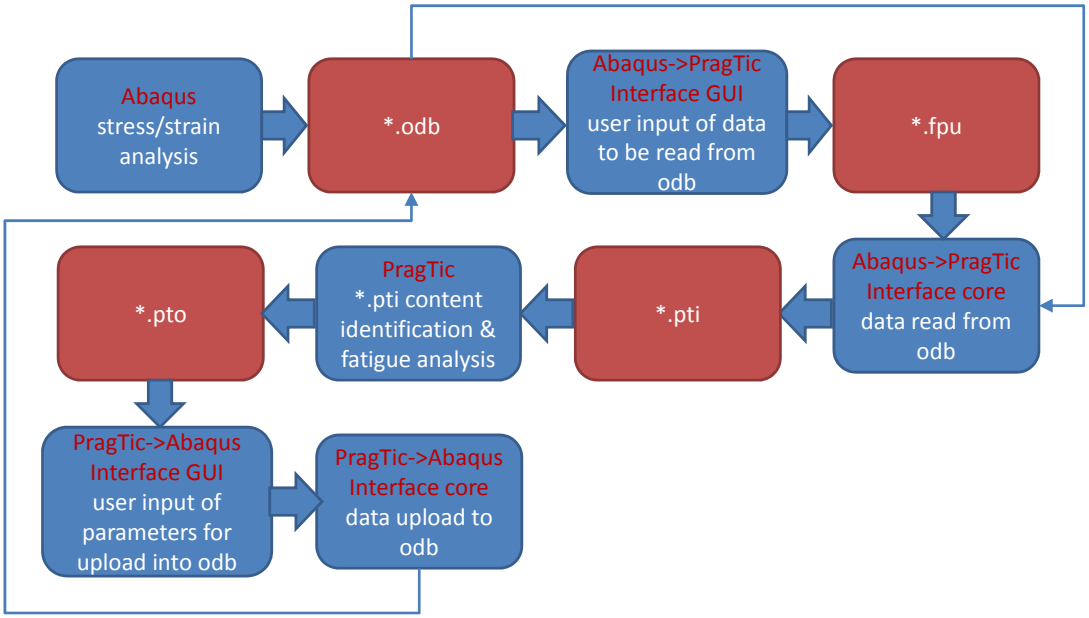

Fig. 1: Chart of data flow between Abaqus and PragTic.

5 Instructions for use of Abaqus2PragTic

The Abaqus2PragTic interface is run within Abaqus/CAE either in GUI or noGUI mode. In this section use of the interface in GUI mode will be presented.

After copying the GUI files into *abaqus_plugins* directory the interface can be executed from *Plug-ins* in the Abaqus main menu bar – Fig. 2. Abaqus/CAE then automatically switches to Visualization module.

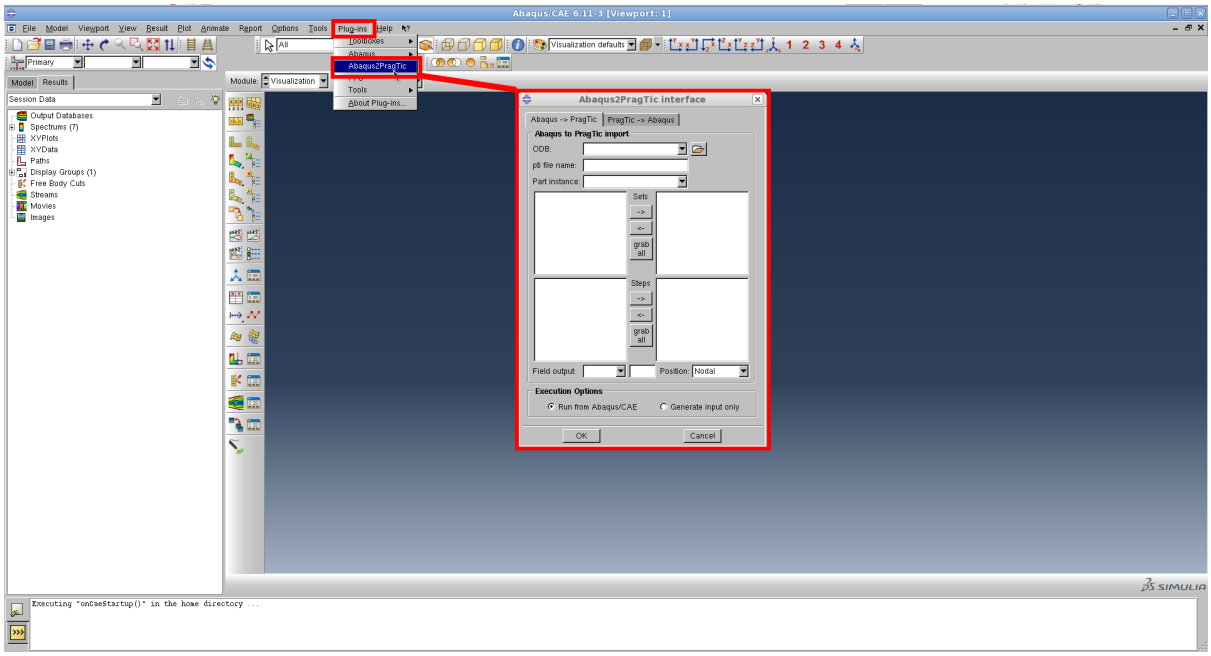

Fig. 2: Basic view of the Abaqus2PragTic interface GUI within Abaqus/CAE.

The main window of the interface contains two tabs in the upper part, which make logical partition of the application according to data export/import direction. The left tab provides interface for exporting data from Abaqus to PragTic (see Fig. 3), while the interface on the right tab (Fig. 4) can help you to move data the other way round.

5.1 Abaqus to PragTic export – GUI mode

The procedure of exporting data from Abaqus to PragTic always starts by selecting an *.odb* database. If one or more databases are currently opened in Abaqus/CAE, the desired one can be selected from *ODB* combo box widget. If no database is opened at the moment, the *open odb database* button can be used for invoking the open file dialogue – Fig. 5.

When the appropriate *.odb* file is selected, the *.pti*[2] file name, which will store the data extracted from *.odb* according to the user input, is automatically created from the *.odb* file name. The *.pti* file will be stored in the same directory as *.odb* database. The content of *.odb* file is analysed and *part instance* combo box, *geometry set* list and *analysis step* list are updated by the recognized items.

The *geometry set* list can contain in the current version of the interface only sets defined on the level of model assembly. The sets are selected and unselected by pushing

[2]PragTic input file

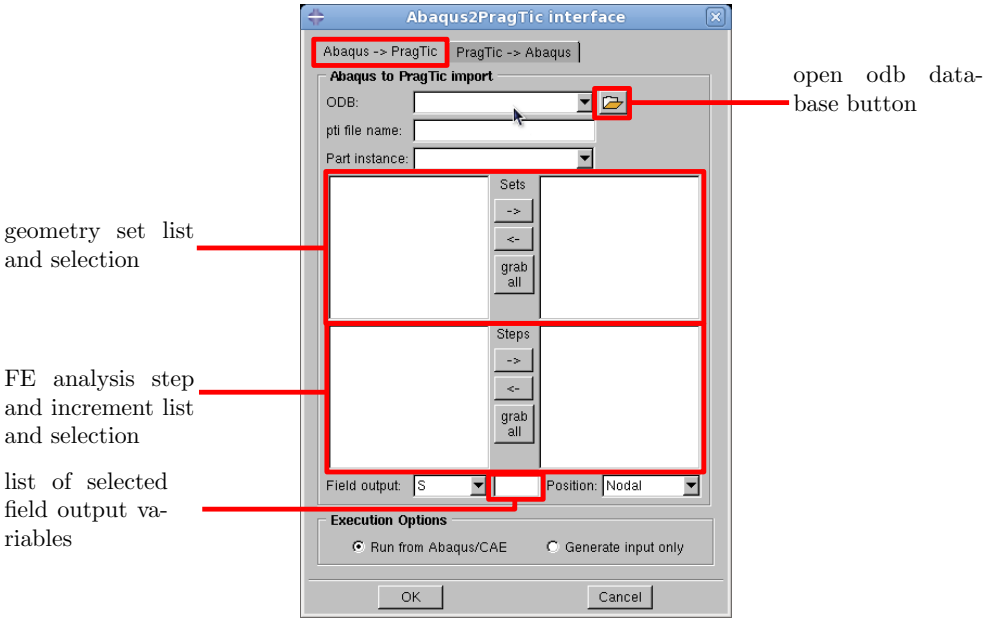

Fig. 3: Detailed view of the Abaqus to PragTic export tab.

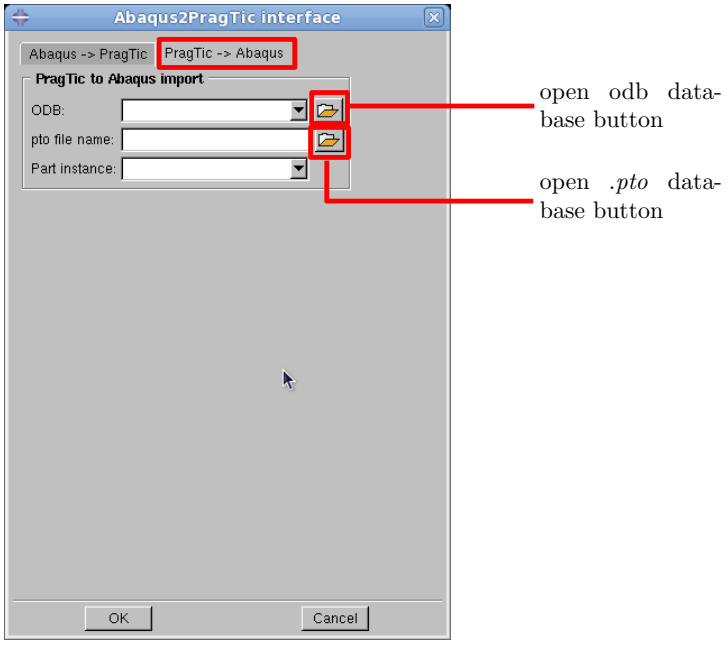

Fig. 4: Detailed view of the Pragtic to Abaqus import tab.

the arrow buttons or *grab all* button. If no set is selected, topology and results of the whole part instance will be stored in *.pti* file.

The *analysis step* list shows all recognized FE analysis step names. Step position number in the analysis step sequence is added to each item and enclosed in squared brackets – Fig. 6. The added numbering can help you with orientation when you compose a load cycle from steps and increments.

When steps are moved from the left to the right list, the increment sequence of each

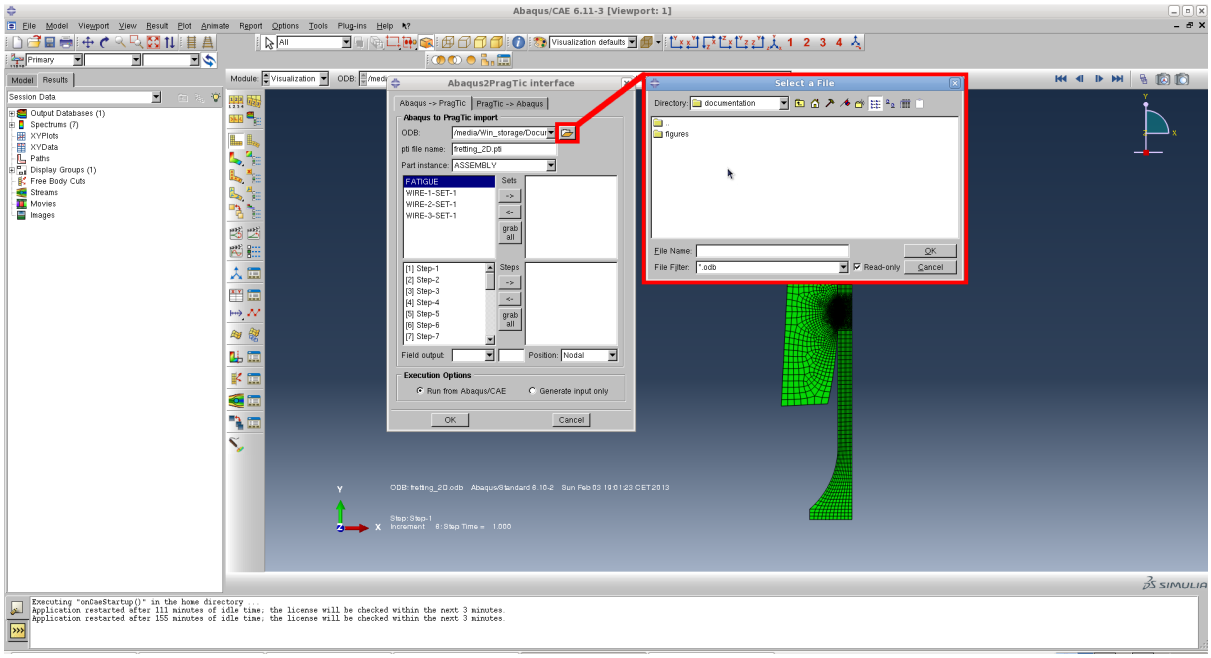

Fig. 5: File open dialogue and state of the interface GUI after opening an *.odb* database.

step is created and appended to each step name as a nested list of items – Fig. 6. The list can be expanded by clicking the *plus sign* and collapsed by *minus sign*, which appear in front of the step name.

The *field output* combo box lists all output variables, which are currently supported by the interface. Currently no checks are done by the program in order to find out if the variables are present in the selected step and increments. The supported field output variables are as follows:

S	components of the stress tensor,
E	total strain components; total strain is composed of elastic, inelastic and thermal strains,
U	displacement vector components,
NT11	nodal temperature.

If one of the output variables is selected in the *field output* combo box, the list of selected output variables is updated by appending the variable to the actual string.

User can specify position from which the field output values will be extracted. Default is *Nodal* position, a complete list of supported positions is as follows:

nodal	averaged values in nodes,
element nodal	non-averaged values in nodes of elements,
integration points	values in element integration points,
centroid	values in element centroid.

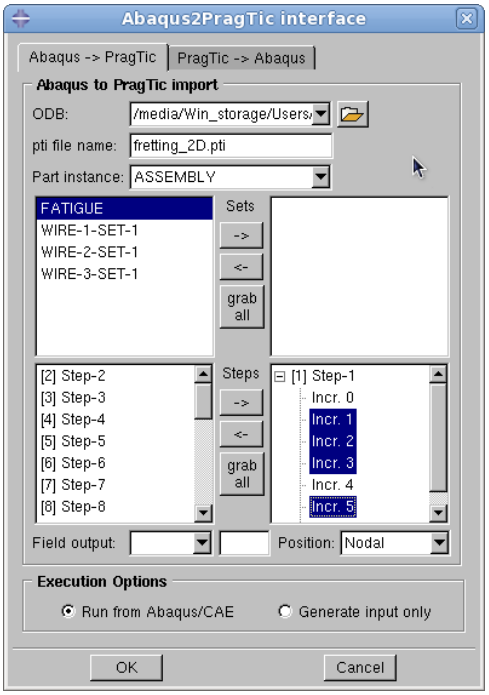

Fig. 6: Analysis step selection. Expanded list of increments under a selected step. The labelled increments can be removed from the list by pushing the left arrow button.

In *execution options* group box user can specify whether the procedure for *.pti* file creation will be run immediately from the opened Abaqus/CAE session or whether only *.fpu* file will be written by selecting "Generate input only". This file stores input provided by user and can be used to run procedure for creating *.pti* file from the command line – section 5.2.

5.2 Executing Abaqus to PragTic export from command line

Command line execution of Abaqus to PragTic export can be very useful in cases where large models are to be transferred. This option bypasses the need for opened Abaqus/CAE GUI during the whole export run time. The export script can be run as a background process on local computer or server.

An *.fpu* text file with user specified details about exported entities has to be provided to the script in *Abaqus_Pragtic_cmdl.py*. The script is executed again within Abaqus/CAE but with "noGUI" option. An example of command follows:

```
[...]$ /usr/abaqus/Commands/abq6113 cae noGUI=
    /home/localuser/abaqus_plugins/Abaqus2PragTic/Abaqus_Pragtic_cmdl.py --
    modelForExport.fpu &
```

.fpu file can be created either by using the interface GUI – section 5.1, or manually in an ASCII text editor. The following rules currently hold for *.fpu* file syntax:

- Data lines beginning with '*' character are considered as comments.

- There are several keywords, which are recognized by the core export scripts. Each keyword has to be provided on a separate text line.
- Keywords have options, which are separated by '=' character without any blanks. If more options are applicable for a given keyword, these options are separated by comma.
- No strict order in keyword input is required except *STEP* keyword. *STEP* keyword order has to guarantee that increments of a single step are listed in a continuous sequence.
- Keywords and options are case sensitive.

The supported keywords are:

ODB	*.odb* database name; full path is required,
	options: string specifying *.odb* file name,
PTI	*.pti* file name without directory path
	options: string specifying *.pti* file name,
INSTANCE	model instance name,
	options: string specifying instance name,
SET	model geometry set name,
	options: string specifying geometry set name,
STEP	step name and appropriate increment number,
	options: step name, increment number,
FIELDOUTPUT	list of field output variables,
	options: comma separated list of field output variables,
POSITION	position of results with respect to element topology,
	options: string specifying result position.

5.3 Importing data from *.pti* file to PragTic

When a *.pti* file with desired data is created, import to PragTic can start. The import *.pti* file dialogue is shown in Fig. 7. The dialogue window is accessed through *File* in the PragTic main menu bar and *Import...*. This command opens a file open dialogue firstly, in which you select the *.pti* file with data to be imported. Contents of the *.pti* file is then automatically analysed and basic statistics about recognized items is shown in the left part of the import dialogue window. You can preselect certain entities only or push *Get All* button to import complete content of the *.pti* file. When *Run* button is pushed, the *.pti* is translated to PragTic database files and fatigue analysis can be performed.

5.4 Exporting fatigue results from PragTic to *.pto* file

PragTic allows you to visualize results of fatigue prediction in Abaqus/Viewer on the source FE model. This can be done by backward export/import procedure by creating

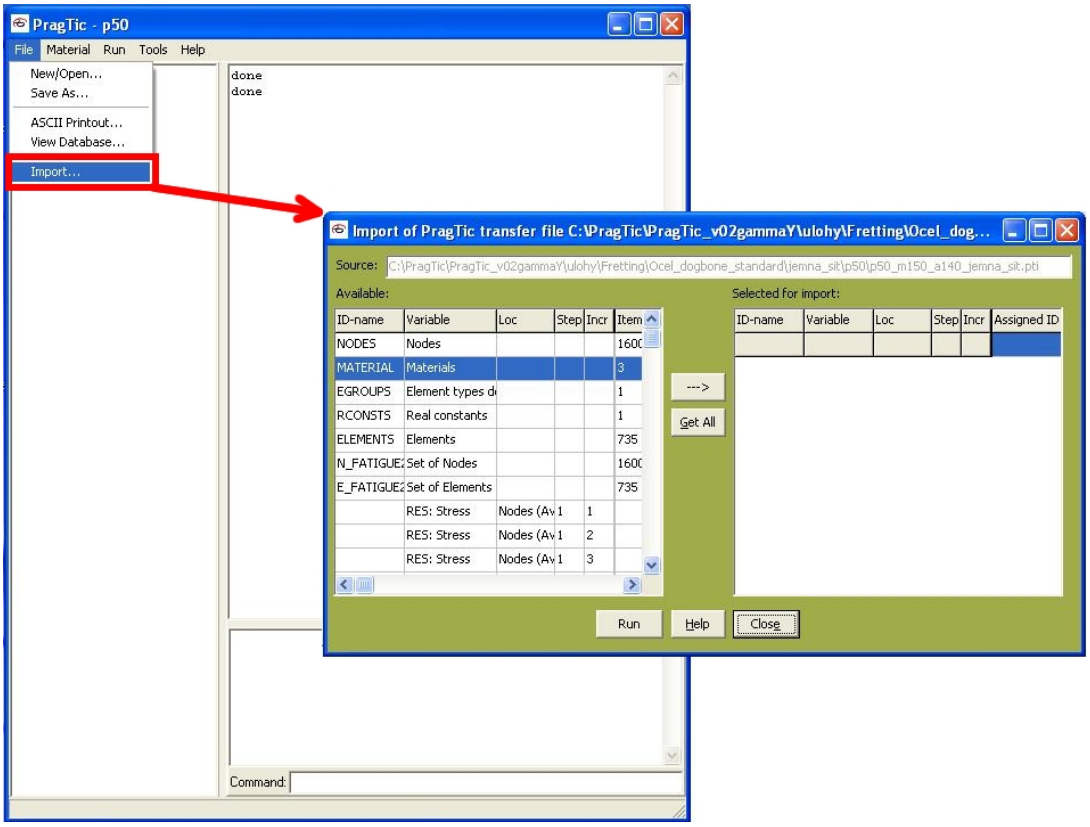

Fig. 7: Dialogue window for *.pti* file data import to PragTic.

.pto file with results from PragTic – Fig. 8. At first, the *.pto* export command asks for *.pto* file name and then the file is created automatically.

5.5 Importing fatigue results from *.pti* file to Abaqus

Import of fatigue results is done via right tab of Abaqus2PragTic interface – Fig. 4. User has to select proper *.odb* database, which stores FE model, *.pto* file with fatigue results and name of the part instance over which fatigue analysis was performed. After pushing *RUN* button content of *.pto* file is translated and saved to specified *.odb*. **It is strongly recommended to make a backup copy of the *.odb* database file before uploading fatigue results.**

In *.odb* fatigue results are referenced as a set of field output variables in a newly created step and frame. Viewing and further processing can be done by standard Abaqus/-Viewer tools.

6 Limitations

There are several limitations which restrict usage of the Abaqus2PragTic interface. Most of them can rather be classified as "application features" since current functionality al-

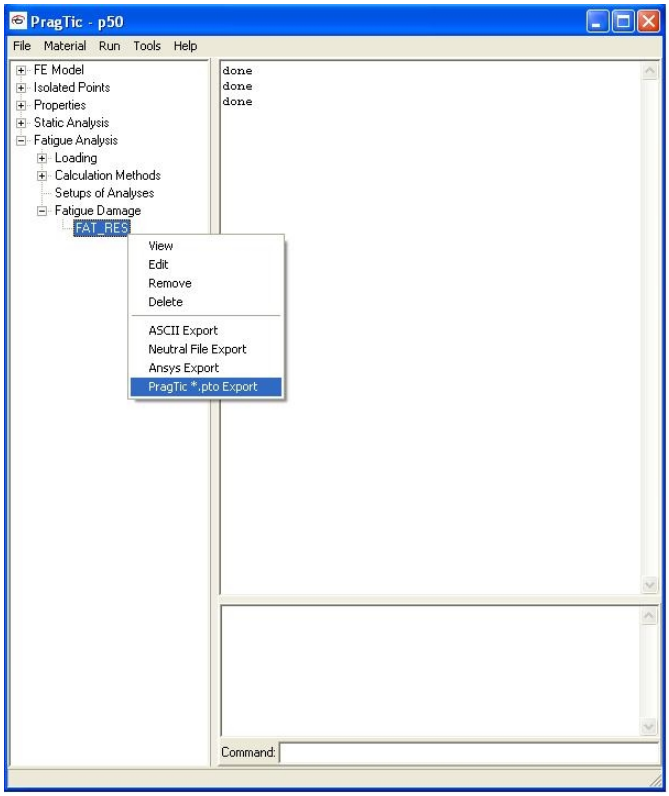

Fig. 8: Command for fatigue results export from PragTic to *.pto* file.

lows you to perform full fatigue analysis and visualization of results. Most likely these limitations will be eliminated in the future. The list of known limitations follows:

1. Only 3D continuum elements are supported.

2. Discontinuous numbering of Abaqus mesh parts created from part assembly will not allow you to create *.pti* file.

3. Only geometry sets defined on a single part instance are supported. This limitation requires that the set was created in Abaqus/CAE during model preparation. Otherwise it can be overcome by creating element set and set of underlying nodes (both with the same label) in *.odb* by using the Abaqus scripting interface – see the Abaqus manual.

4. Command line import of *.pto* data to *.odb* is not supported. This operation can be run only within Abaqus/CAE GUI. But you should be aware that in general it takes only a small fraction of time comparing to the data export from *.odb* to *.pti*, i.e. backward procedure.

5. Currently name of the new step, which is created during import of *.pto* data to *.odb* is generated by PragTic according to the result set name. It can cause conflicts once you have more PragTic databases which deal with fatigue analysis of the same

FE model. In such cases, you can set an optional step name by manual change of this parameter in *.pto* file.

7 Acknowledgement

The authors would like to acknowledge support from the Grant Agency of the Czech Technical University in Prague, grant No. SGS12/175/OHK2/3T/12, and support from Technology Agency of the Czech Republic, grant No. TA01011274.