MpCCI
FSIMapper

Part I

—

MpCCI FSIMapper

Version 4.8.0

MpCCI 4.8.0-1 Documentation
Part I   FSIMapper
PDF version
March 20, 2025

MpCCI is a registered trademark of Fraunhofer SCAI
www.mpcci.de

Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Schloss Birlinghoven 1, 53757 Sankt Augustin, Germany
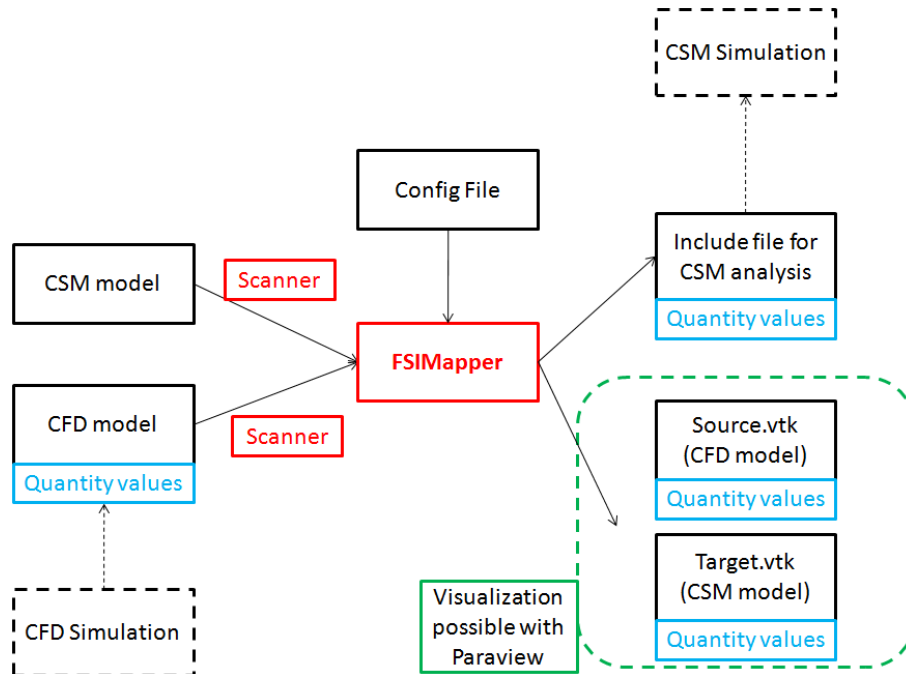
1 MpCCI FSIMapper Overview

Figure 1: Software architecture of the MpCCI FSIMapper in batch usage (cf. section 6)

Mapping

For many problems the influence of the CFD solution on the CSM solution is a lot more remarkable and significant than vice versa. If a one-way connection between the fluid and the solid solver shall be established, quantity values – namely temperature or pressure values that were calculated as a part of the CFD solution – have to be transferred from the CFD to the FEM mesh. Then the structural mechanics solver can use these values as a boundary condition. At the interface between the fluid and solid domain the two meshes usually do not match and interpolation of quantities becomes necessary. This transfer of quantity values from one mesh to another can be done easily with the MpCCI FSIMapper. While "one–way" fluid–structure interactions certainly are the main application of the MpCCI FSIMapper, the tool can be used generally to map values from one mesh to another.

Supported software

At the moment the MpCCI FSIMapper is able to read FLUENT (.cas and .dat, .cas.h5 and .dat.h5), CFX (.csv), MentorGraphics FloTHERM (.flofea, .csv), MentorGraphics FloTHERM XT (.txt), MentorGraphics FloEFD (.efdfea), MagNet (.vtk), Fidelity Fine Turbo (.cgns), MSC Nastran Bulk (.bdf, .nas, .dat), Abaqus (.inp), ANSYS Maxwell (*.unv) and EnSight Gold Case as source result files. Since the EnSight Gold Case export is also supported by CFX, FLUENT, ANSYS Icepak, STAR-CD, STAR-CCM+, etc. , so these simulation codes are supported indirectly.

Supported native CSM codes are Abaqus (.inp) input decks, ANSYS (.db and .cdb), MSC Nastran Bulk (.bdf) data input files, LS-DYNA keyword (.key, .k, .dynain) and also neutral EnSight Gold Case file format is supported.

Supported meshes and quantities

The two meshes between the interpolation shall take place have to be 2–dimensional surface or 3–dimensional volume meshes in the 3–dimensional space. For Abaqus and MSC Nastran also 1–dimensional surfaces of 2–dimensional meshes are supported.

The physical quantities that can be read and mapped are temperature, wall heat transfer coefficient, the wall heat flux, pressure and force density. Furthermore, it is possible to compare two geometries with the MpCCI FSIMapper.

MpCCI FSIMapper output files

For Abaqus, ANSYS, MSC Nastran and LS-DYNA mapped quantity values can be written in native solver format, e.g. Abaqus input, ANSYS APDL, MSC Nastran Bulk and LS-DYNA keyword format respectively, that can be easily included in the original CSM input deck. In addition to native format, the MpCCI FSIMapper allows export to neutral EnSight Gold Case format which is supported by a wide range of simulation software.

Additionally, the MpCCI FSIMapper saves the source and the target model in the native .ccvx format of MpCCI Visualizer. In this way the mesh and data used for and resulting of the mapping can be visualized afterwards.

For mapped harmonic data or Fourier transformed transient data the MpCCI FSIMapper exports a .ccvx file where the complex excitations are transferred back to the time domain (per frequency). Thus, the real and imaginary part of the excitations can be visualized as transient fluctuations (using pseudo time steps).

Supported operating systems

The tool is available for Windows and Linux for 64–bit machines in graphical and batch mode.

MpCCI FSIMapper concept

The MpCCI FSIMapper comes with a graphical user interface where all relevant parameters for the mapping can be set. The usage of the tool is quite simple: after selecting the models and parts some choices concerning algorithms and parameters have to be made. After the mapping the results and the original values are loaded into the MpCCI Visualizer.

The immediate visualization of the results makes it possible to detect model errors or bad mapping results due to parameter problems. The user can identify the critical areas and start a new mapping process.

Additionally, the MpCCI FSIMapper can be used as a batch tool. The parameters and settings that are normally made in the GUI can be defined in a simple ASCII configuration file.

2 MpCCI FSIMapper Installation

The installation of the MpCCI FSIMapper is quite easy. It may be either part of the MpCCI installation or a standalone version.

2.1 Part of MpCCI Installation

As part of the MpCCI installation the MpCCI FSIMapper is completely integrated into the MpCCI environment.

No further actions have to be done.

2.2 Standalone Version

The installation of the standalone version is based on the following steps:

- The MpCCI FSIMapper has to be installed.
- Perl has to be installed.
- A license for the MpCCI FSIMapper must be available.

There are two ways to install the MpCCI FSIMapper:

- via the *Windows installer* for a local Microsoft Windows only installation and
- via the *multi-platform distribution file* for Linux and file server installations.

A Microsoft Windows installation requires a free disc space of approx. 50 MB and a multi-platform installation of approx. 100 MB (without tutorials).

2.2.1 Local Microsoft Windows Installation

For a local Microsoft Windows installation you download an "MSI" file.

Execute this installer file with administrative rights and follow the instructions. The installer unpacks the MpCCI FSIMapper for all users, does registry entries and adds firewall rules which are necessary for the communication between the MpCCI FSIMapper and the MpCCI Visualizer. It also checks if Perl is installed. If no Perl executable could be found in the path, the user is offered an Open button to download Strawberry Perl. This installation also takes place via a Microsoft Windows installer ("MSI" file).

The MpCCI FSIMapper software is now installed on your system.

For running the MpCCI FSIMapper a license must be available (see ▷ 2.2.4 License ◁).

If you did not install Perl or if your Perl installation is too old see ▷ 2.2.3 Perl ◁.

2.2.2 Linux or Multi-Platform Installation

If you intend to install MpCCI FSIMapper under Linux or on a file server with access for Linux and Microsoft Windows users you should have downloaded a multi-platform distribution file.

The installation is just done by unpacking the downloaded archive to a folder of your choice. Then the PATH environment variable needs to be extended by the "bin" directory of the MpCCI FSIMapper. For Linux everything can be done without any administrator privileges if you use a local installation directory. For Windows administrative rights are necessary to add firewall rules.

The steps in detail:

1. Unpack the downloaded archive on Linux:
   - Change to your installation directory e.g. `cd <your home>/software`.
   - Move the downloaded file to your installation directory
   - Extract the MpCCI FSIMapper files from the downloaded file with the command
     `tar zxvf mpccifsimapper-<FSIMapper-version>.tar.gz`
     or, if the z option is not available, with the commands
     `gunzip mpccifsimapper-<FSIMapper-version>.tar.gz`
     `tar xvf mpccifsimapper-<FSIMapper-version>.tar`

2. Update your user PATH environment with the MpCCI FSIMapper "bin" directory:

   Linux: `export PATH=<your install dir>/FSIMapper/bin:$PATH` for korn shell (ksh) and bash resp.
   `setenv PATH <your install dir>/FSIMapper/bin:$PATH` for c-shell (csh) and tc-shell (tcsh) users.

   Windows: In the Windows Control Panel go to where you can change own user environment variables. Extend the PATH environment variable by `<your install dir>\FSIMapper\bin`.

3. Windows only: Add firewall rules for MpCCI FSIMapper executables.
   In the Control Panel → System and Security → Windows Firewall select Allow a program or feature through Windows Firewall. Now click on Allow another program... and add
   - "`<your install dir>\FSIMapper\bin\windows_x64\mpcci_fsimapper.exe`" and also add
   - "`<your install dir>\FSIMapper\bin\windows_x64\mpcci_visualizer.exe`" and finally add
   - "`<your install dir>\FSIMapper\bin\windows_x64\mpcci_ccvxcat.exe`"

The MpCCI FSIMapper software is now installed on your system.

For running the MpCCI FSIMapper a license must be available (see ▷ 2.2.4 License ◁).

If you run into problems with Perl while executing the MpCCI FSIMapper have a look at ▷ 2.2.3 Perl ◁.

2.2.3 Perl

Perl is a platform independent script language which allows running identical scripts under Linux as well as Microsoft Windows. The MpCCI FSIMapper uses perl scripts for executing its commands. Therefor it requires a working Perl installation. If Perl is already installed on your system - this is true for nearly any Linux system - you may check the Perl version by typing

```
> perl -version
```

For MpCCI FSIMapper under Linux you need at least Perl 5.6. Under Microsoft Windows you can either use Strawberry Perl or ActivePerl from version 5.8 up.

If Perl is not installed or if the version is too old, please contact your system administrator for installation resp. upgrade. On Linux systems it's part of the distribution but Perl is never part of your Microsoft Windows system. For Microsoft Windows it can be downloaded from the world wide web. We recommend to have either Strawberry Perl or ActivePerl from version 5.8 up installed.

Strawberry Perl is free of charge and can be downloaded from strawberryperl.com/. A 64 bit version of Strawberry Perl for Windows 64 bit is available.

ActivePerl is covered by the ActiveState Community License and can be downloaded from www.activestate.com/activeperl.

(!) Perl is not part of the MpCCI FSIMapper installation. Only for the local Microsoft Windows installation (▷ 2.2.1 Local Microsoft Windows Installation ◁) the download of the third party software Strawberry Perl is integrated. The installation of Strawberry Perl should be done as the next step.

2.2.4 License

2.2.4.1 Installing the MᴘCCI License Server

MpCCI FSIMapper uses a FlexNet Publisher based floating license mechanism. FlexNet Publisher license server has to be started on the license server host defined in the MpCCI FSIMapper license file (see ▷ 2.2.4.2 Requesting a License ◁). You can run MpCCI FSIMapper anywhere in your internal network. For more information about FlexNet Publisher, please refer to the FlexNet Publisher license administration guide.

You will find the software packages for the MpCCI license server installation (FlexNet Publisher tools and vendor daemons for SVD) in the SCAI download area (www.mpcci.de → MpCCI Download Area → enter your username and password and login → License Tools). Please select the suitable platform and execute the installer for Microsoft Windows resp. unpack the downloaded package for Linux.

In general FlexNet Publisher comes with some tools (lmutil, lmstat, lmhostid, lmgrd, lmdown etc.) for managing the licenses - please refer to the FlexNet Publisher documentation.

MpCCI FSIMapper only needs three FlexNet Publisher executables:

the utility:	lmutil
the license server:	lmgrd
the vendor daemon:	SVD

The MpCCI FSIMapper vendor daemon is named SVD. The FlexNet Publisher port number of the SVD vendor daemon is 47000. If there are other software packages installed using also FlexNet Publisher there will be several FlexNet Publisher utils available. Depending on your local installation and your own PATH environment it is not always defined which of these lmutils will be executed upon a command call.

2.2.4.2 Requesting a License

You need to acquire a license file for the MpCCI FSIMapper from Fraunhofer SCAI.

Therefor you need to know the hostname and the hostid of the license server:

Please login on the host where the FlexNet Publisher license server for MpCCI FSIMapper should run on. In the following example, "> " is your prompt:

```
> hostname
myHostName

> lmutil lmhostid -n
12345abcd
```

If you have multiple network devices installed (ethernet card, wireless LAN, docking station on a notebook) you may see multiple hostids. For the MpCCI FSIMapper license daemon the integrated ethernet card address should be the correct one.

Please go to the download area at www.mpcci.de to get an MpCCI FSIMapper license file.

If you already have an account for the download area the MpCCI FSIMapper license file can be requested via the *License Request Form Sheet*:

1. At www.mpcci.de go to `Download Area`.

2. Click the link `Log into the MpCCI Download Area` which will redirect you to the Fraunhofer SCAI download portal.

3. Enter your account details in the E-Mail and Password input fields.

4. Go to the menu **MpCCI** and select your product:
   `MpCCI FSIMapper`.

5. Then click `License Request`.

Please fill in the form with all required data - including the hostname and hostid of the license server (**myHostName** and `12345abcd` in the example above) - select **FSIMapper** in the **Mapper** section, leave selections empty for **MpCCI Jobs**, **Parallel Clients**, **Code Adapter** and **AddOns** and submit. You will receive the required license file soon after via e-mail.

Please copy your received license file into the file "<*fsimapper_home*>`/license/mpcci_fsimapper.lic`".

Then, please start the **FlexNet Publisher** license server daemon and SVD vendor daemon on the local host with the following command:

```
> lmgrd -c <LICENSE_FILE>
```

In the case of problems you can get information about the license server in a logfile with the command:

```
> lmgrd -c <LICENSE_FILE> -l <LICENSE_LOGFILE>
```

To stop the license server please type:

```
> lmutil lmdown -c <LICENSE_FILE>
```

You should set the generic **FlexNet Publisher** variable `SVD_LICENSE_FILE`.

For ksh or bash users:

```
> export SVD_LICENSE_FILE=47000@<HOSTNAME>
```

For csh users:

```
> setenv SVD_LICENSE_FILE 47000@<HOSTNAME>
```

For **Microsoft Windows** users:

```
> set SVD_LICENSE_FILE=47000@<HOSTNAME>
```

This variable is also referred to by other software packages using **FlexNet Publisher**;

`SVD_LICENSE_FILE` may comprise several entries $< port@host >$ (one for each application). If there is more than one entry they must be separated by a ":" on **Linux** and by a ";" on **Microsoft Windows**.

We suggest that you set the `SVD_LICENSE_FILE` environment variable in your login file (`".cshrc"` or `".profile"` or `".bashrc"`) for **Linux** users resp. for **Microsoft Windows** users in the **Environment Variables** window of the system settings.

FlexNet Publisher stores data of a previous successful connection to a license server in the file `"<HOME>/.flexlmrc"`. You may create or adjust this file.

After setting this environment variable you can check whether the license server is running and can be reached from the machine where you are currently working and from where you plan to start **MpCCI FSIMapper**:

```
> lmutil lmstat -vendor SVD
```

For more details command

```
> lmutil lmstat -a -c <LICENSE_FILE>
```

### 2.2.5 Configure a License Manager as UNIX Service

On UNIX, edit the appropriate boot script, which may be `"/etc/rc.boot"`, `"/etc/rc.local"`, `"/etc/rc2.d/Sxxx"`, `"/sbin/rc2.d/Sxxxx"`, etc. Include commands similar to the following.

```
<LICENSE_TOOL_INSTALLATION>/bin/lmgrd -c <LICENSE_FILE> -l <LICENSE_LOG_FILE>
```

The LICENSE_TOOL_INSTALLATION is the full path of the license software installation. This command will create a log file under `"LICENSE_LOG_FILE"` and you have to ensure that the process could write in the directory.

2.2.6 Configure a License Manager as Windows Service

Execute the `"lmtools.exe"` application from the license manager installation directory: `"<LICENSE_TOOL_INSTALLATION>/bin/lmtools.exe"`
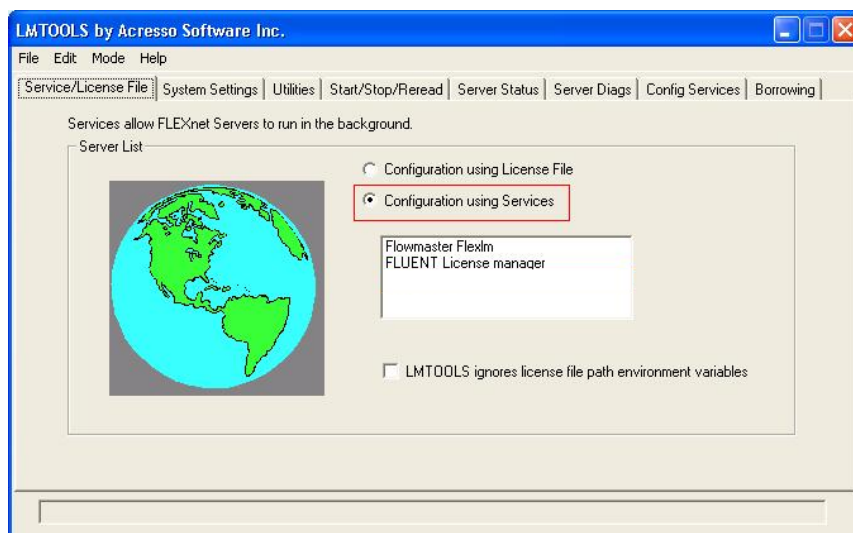
- Select in the Service/License File tab section the option Configuration using Services.

- Click the Config Services tab section.
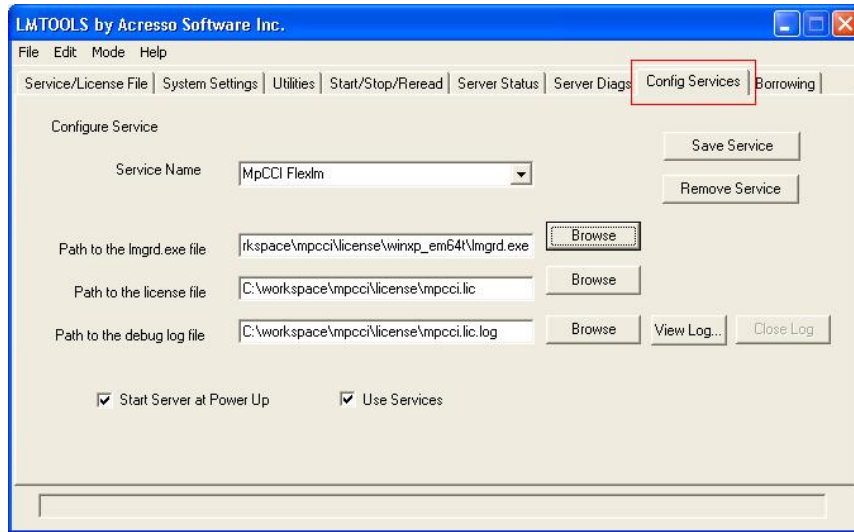
- Enter a service name e.g. MpCCI license manager or MpCCI FLEXlm.
- Select the path of the program "lmgrd.exe" with the Browse button:
  "<LICENSE_TOOL_INSTALLATION>/bin/lmgrd.exe"
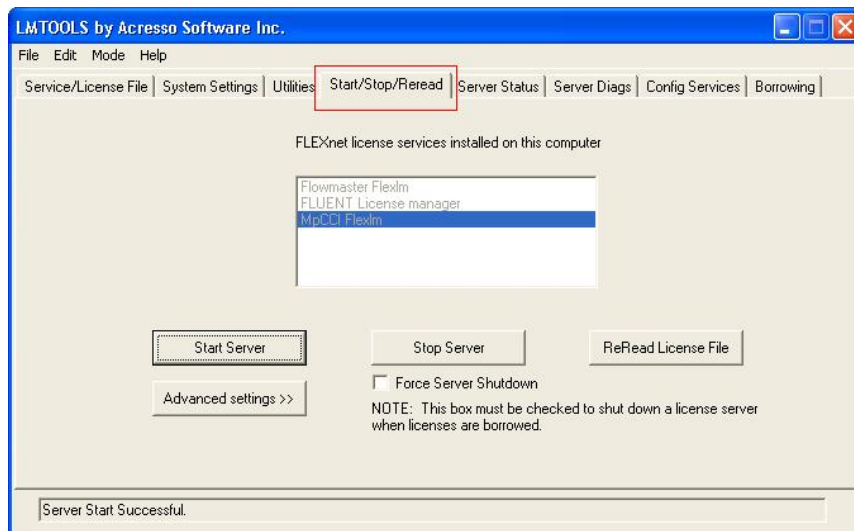- Select the license file "mpcci.lic" with the Browse button:
  "<LICENSE_TOOL_INSTALLATION>/license/mpcci.lic"
- Activate the Start Server at Power Up option.
- Activate the Use Services option.
- You can optionally add a log file by providing a file name for the Path to the debug log file option.
- Click on the Save Service button.

- Select in the Start/Stop/Reread tab section the license service.
- Click on the Start Server button.
- The license server is now running and configured to start at power up.

3 MpCCI FSIMapper Command

```
Usage:
   fsimapper [-]option


Synopsis:
   'fsimapper' is used to launch the MpCCI FSIMapper.


Options:
   -batch  <configFile> <source> [source_quant] <target>
        Use this option to launch the MpCCI FSIMapper in batch mode.
        Provide a configuration file, source and target model files and
        an optional source quantity file.

   -help
        This screen.

   -scan  <FORMAT> <model>
        Use this option to scan models with the MpCCI FSIMapper.
        Supported scanner formats are ABAQUS, ANSYS, LSDYNA, FLUENT,
        NASTRAN, CFXCSV, FLOTHERMMAPLIB, FLOEFDMAPLIB, FLOTHERMXT,
        FINETURBO, MAXWELL, 6SIGMAET, VMAP  and ENSIGHT.
        The scanner output will be written to stdout.

   -convert <FORMAT> <fileToConvert> [releaseNumber] [ansysProduct]
        Use this option to convert an input file of special format
        into own intermediate format '.ml'.
        At this time only ANSYS '.cmd' and '.db' files are supported
        to be converted so FORMAT must be ANSYS.
        Optional arguments are the release number for the used ANSYS
        executable and the ANSYS product to use for starting ANSYS.
        By default the latest ANSYS release will be taken.

   -visualize <ccvx-File-1> <ccvx-File-2> ...
        Opens MpCCI Visualizer and loads all passed CCVX files
```

The batch usage is described in ▷ 6 Batch Usage of the MpCCI FSIMapper ◁ the scan usage in ▷ 6.1 File Scanners ◁ and the convert usage in ▷ 5.2.4 ANSYS Scanner and Converter ◁.

The MpCCI FSIMapper itself is described in ▷ 4 MpCCI FSIMapper GUI ◁.

4 MpCCI FSIMapper GUI

This section describes the usage of the MpCCI FSIMapper. The different panels of the graphical user interface are described in the next sections. Details on the used mapping and orphan filling parameters can be found in ▷ 7.1 Numerical Methods ◁.

4.1 Starting the MpCCI FSIMapper

The MpCCI FSIMapper GUI is embedded in the MpCCI Visualizer and can be started from command line by

```
> fsimapper
```

launching the MpCCI Visualizer with activated plugin for file based mapping. To open the MpCCI FSIMapper GUI press button ▨ after the MpCCI Visualizer has started.
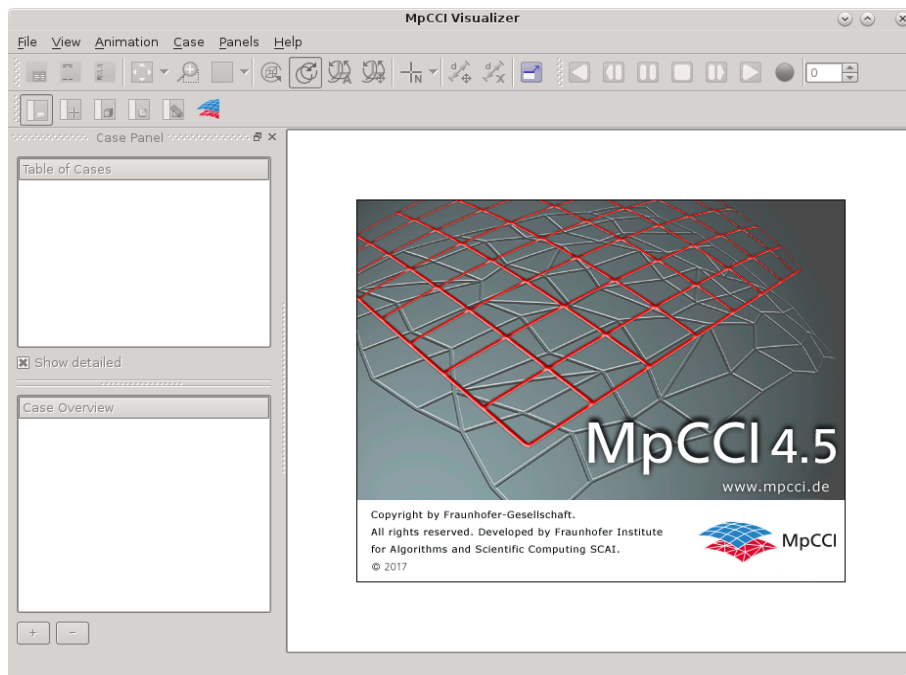

Figure 2: The MpCCI FSIMapper plugin location in the MpCCI Visualizer

The MpCCI FSIMapper writes a configuration setup file containing user model and part selection and launches a separate process which executes the defined tasks. When finished the models together with quantities are being sent to the MpCCI Visualizer where each model appears in a separate viewport.

4.2 The "What to map" Panel
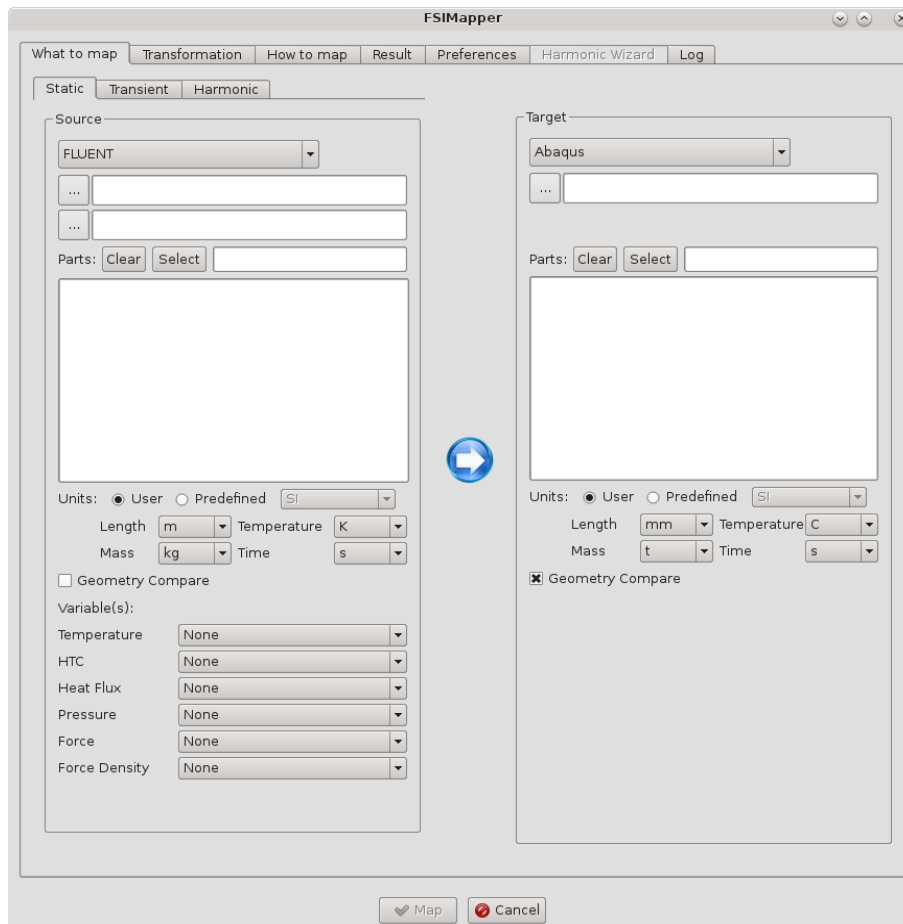
Figure 3: The "What to map" panel of the MpCCI FSIMapper

4.2.1 Selection of Cases, Parts and Quantities

In the "What to map" panel of the mapping dialog, at first the type of analysis needs to be specified. The user can switch between "Static", "Transient" and "Harmonic". Using the "Static" panel means to map one state from a source model to a target model. For the "Transient" panel it is assumed that the data is present for several time steps.

Additionally to the creation of transient loading, there is the possibility to perform a Fourier transformation on the transient data in order to do a frequency response analysis, cf. ▷ 4.5.2 Apply Fourier Transformation ◁.

In the "Harmonic" panel, complex loading is assumed in order to map excitation loads for NVH analyses.

At second the source and target models need to be specified. The file type for the two models can be selected in a drop-down list. The files can be selected with the open file dialog.

Source File Specification

For a "Static" source model FLUENT, Mentor Graphic's FloTHERM, FloTHERM XT and FloEFD, CFX, Fidelity Fine Turbo, Infolytica's MagNet and EnSight Gold Case are supported. Moreover, the MSC Nastran bulk data format and the Abaqus input format is supported.

For a "Transient" source model MSC Nastran, MagNet, FloTHERM, ANSYS Maxwell and EnSight Gold are supported. The target model can be either in Abaqus, ANSYS, LS-DYNA or MSC Nastran format. Currently only the "Force", "Force Density" and "Pressure" variables are supported for transient analyses.

For a "Harmonic" source model Fidelity Fine Turbo and EnSight Gold are supported for the "Pressure" variable. The target model can be either in Abaqus, ANSYS or MSC Nastran format.

(!) As harmonic CFD simulations often result in a set of harmonic quantities, the "Harmonic Wizard" offers the possibility to map all or a subset of the available harmonics at once. Refer to ▷ 4.7 The "Harmonic Wizard" Panel ◁.

(!) For transient FloTHERM *.csv format as target code only Abaqus is currently supported.

Depending on the selected source code, the following files are required:

- **FLUENT** : the native geometry (∗.cas, ∗.cas.h5) and quantity (∗.dat, ∗.dat.h5) file

- **FloTHERM** : resp. **FloEFD** (∗.flofea, ∗.csv) resp. (∗.efdfea) file exported by the code

- **FloTHERM XT**: (∗.txt) file exported by the code

- **EnSight Gold** : (∗.case) file which references the binary (∗.geom) file and the binary quantity file(s)

- **MagNet** : the Infolytica "MpCCI Exporter" format ∗.vtk, where multiple files can be selected (defining multiple parts)

- **CFX** : the "CFD-Post Generic Export" format (∗.csv)

- **Fidelity Fine Turbo** : the native (∗.cgns) file and optionally the (∗.run) file

- **Abaqus** : ASCII file in the native input file format (∗.inp)

- **MSC Nastran** : ASCII file in the native bulk data format

- **ANSYS Maxwell**: the Maxwell export of geometry (∗.unv) and quantity (∗.unv) file

Target File Specification

The target model can be either in Abaqus, ANSYS, MSC Nastran, LS-DYNA or EnSight Gold Case format. Besides ASCII files in Abaqus input format, MSC Nastran bulk data format or ANSYS format, the binary ANSYS format .db and the binary EnSight Gold Case is supported.

Parts Selection

After the selection of the models, their files are scanned automatically and the available model parts are listed in the selection area below. For better distinction of area types the MpCCI FSIMapper lists each "part" by leading 'V' for volumetric parts, 'S' for surfaces or 'E' for element sets (Abaqus only). By clicking on its name, relevant "parts" for the mapping can be selected. In the most common case this is the interface between the fluid and a solid in fluid-structure interaction.

(!) Either homogen volumes, surfaces or element sets can be selected for mapping. A composition of different area types is not supported and has to be mapped separately.

(i) The "parts" selection can be performed by using the string matching function.
Provide the string to match in the input field and click on the button Select to activate the selection.
By clicking on the button Clear it will clear the current selection of "parts".

Unit Systems

The next step in the GUI is to define the corresponding unit systems for the source and the target meshes. Figure 4 illustrates the part in the "What to map" panel where length, mass, temperature and time units can be defined for both models independently. This is useful when the source and target model length units for the model geometry or the quantity units do not match.

(!) All mapped and displayed quantities are converted and exported due to the defined target unit system.
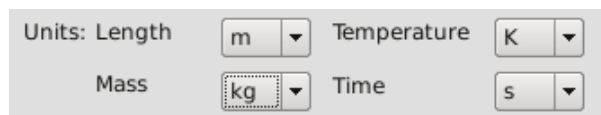

Figure 4: Overview of unit system configuration in graphical user interface

4.2.2 Geometry Compare

For the source and the target models it is possible to tick a check box labeled "Geometry Compare". If this option is selected for the source or target model, additional geometric information is computed and the MpCCI Visualizer receives the two additional quantities `NODAL_DISTANCE` and `ASSOCIATION_DISTANCE`. If only the geometry of the models shall be compared without mapping quantities, this option has to be selected for both models.
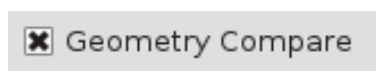
Figure 5: Geometry Compare check box

For each node of one mesh the `NODAL_DISTANCE` is closest geometric distance to a point of the other mesh. However the `ASSOCIATION_DISTANCE` denotes the distance between a point of a mesh and the closest (associated) element surface of the other mesh. If the point lies "in" an element, this would mean that the association distance is 0.

4.2.3 Quantity Identification for CSM Solver Output

As the notation of a physical quantity may differ among native solvers file formats, the user has to specify the physical quantity by its name so that it can be used for export to either Abaqus input, MSC Nastran Bulk files or ANSYS Mechanical APDL files. A selection of the relevant quantities can be done in the lower left part of the "What to map" panel, cf. Figure 6.

(a) "Static" panel   (b) "Transient" panel   (c) "Harmonic" panel

Figure 6: Quantity selection area in graphical user interface

For each of the supported physical quantities "Temperature", "HTC" (Heat Transfer Coefficient), "Heat Flux", "Pressure", "Force" and "Force Density" there exists a drop-down menu where one of the quantity ids of the source file can be selected.
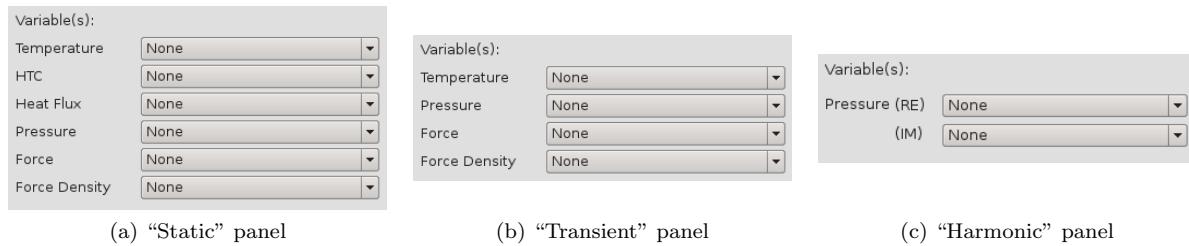
In some cases of the harmonic mapping it is obligatory to specify both real and imaginary part of the excitation quantity, cf. ▷ 4.3 The "Transformation" Panel ◁.
Having in mind that a complex excitation amplitude can be reformulated as amplitude and phase lag of a periodic fluctuation (Euler's formula), it is more realistic to specify both the real and imaginary part of the quantity. If only the real part or only the imaginary part is given, all excitations will vibrate exactly in-phase or out-of-phase (only 0° or 180° phase shift). In this case the vibration amplitudes are the absolute values of the given data.

4.2.4 Execute a Mapping Process

If the actual configuration inside the "What to map" panel is in a valid state, pressing the "Map" button - which is available on all panels - starts the mapping process . A valid mapping state is present if the following prerequisites are satisfied:

- A source model file is specified
- A source quantity file is specified (FLUENT, MSC Nastran, Abaqus and ANSYS Maxwell only)
- A target model file is specified
- At least one part of the source model is selected
- At least one part of the target model is selected
- Quantity selection dependent on the analysis type
  Static and Transient panel
       At least one physical quantity selection is made OR both geometry compare boxes are checked
  Harmonic panel
       Depending on cyclic symmetry:
    − **none**
       At least one physical quantity selection is made OR both geometry compare boxes are checked
    − **cyclic symmetric**
       Depending on data periodicity (cf. ▷ 4.3 The "Transformation" Panel ◁):
       ∗ *constant/alternating*
          At least one physical quantity selection is made OR both geometry compare boxes are checked
       ∗ *paired*
          At least both real and imaginary parts of one physical quantity selection are made OR both geometry compare boxes are checked

If the current state is valid the "Map" button gets available otherwise it is disabled.

4.3 The "Transformation" Panel

4.3.1 The "Geometry" Subpanel
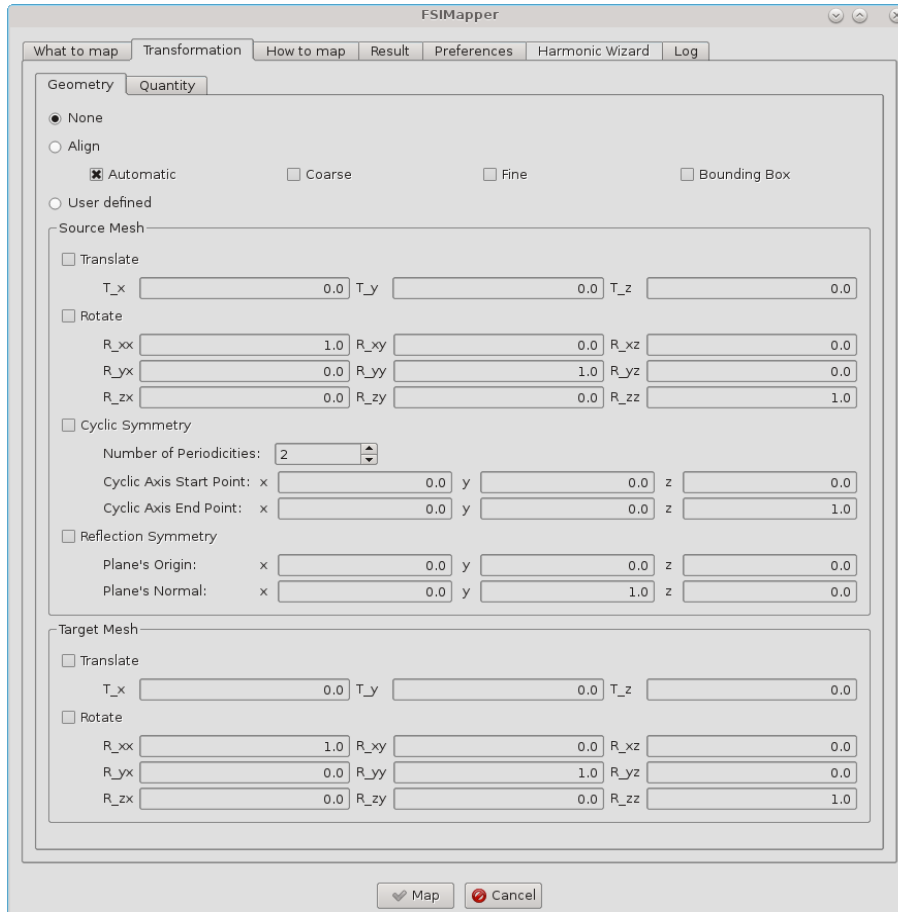
Figure 7: The "Geometry" subpanel of the MpCCI FSIMapper

A common problem in one-way one-step coupling is the use of different local coordinate systems for CFD and CSM simulation. Here the models might not have the same spatial location in a common coordinate system or even represent only a symmetric part of the target model. Hence, for a position-dependent neighborhood computation, both models need to be aligned in a preprocessing step. This alignment step can be done in the "Transformation" panel (Figure 7) where different options for mesh positioning can be selected in the "Geometry" subpanel.

None
> No mesh positioning is needed.

Align
> Select one of the four different automatic alignment options:

> **Automatic**
>> The MpCCI FSIMapper first applies a "Coarse" transformation followed by a "Fine" transformation on the source mesh to move it close to the target mesh location.

Coarse

The MpCCI FSIMapper computes the center of gravity of both models and translates the source towards the target. Then the principal axis of the models will be aligned.

Fine

The MpCCI FSIMapper minimizes the global total nodal distance from the source to the target model.

Bounding Box

The MpCCI FSIMapper computes the global expansion of the models and then translates the source model bounding box to the target one.

User defined

As an automatic alignment might fail in case of symmetric geometry or partial overlap of models the MpCCI FSIMapper also offers an alignment by user defined transformation.

Translate and Rotate

Here for each model the user can define a 3x3 rotation matrix R as well as a translation vector T so that new model coordinates x' are computed as

$$x' = Rx + T.$$

Only a source mesh gets transformed into the target mesh coordinate system

$$x' = R_t^{-1} R_s x + R_t^{-1}(T_s - T_t)$$

where the lower index s denotes source and t target transformation.

Cyclic Symmetry

This transformation provides for cyclic symmetric source models the corresponding full mesh and quantity. In this way a mapping to a full target model or to a periodic target model with a different section shape is possible, as shown in Figure 8. The user needs to define the number of sections as Number of Periodicities and two points (corresponding to the source model position) which define the axis of cyclic symmetry (Cyclic Axis Start Point and Cyclic Axis End Point).
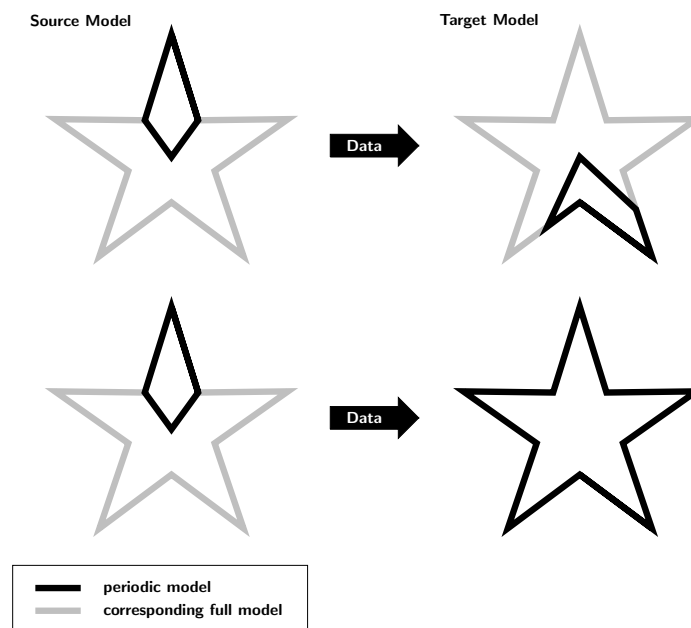

Figure 8: Mapping of data between different periodic sections (black lines) which represent the same full model (grey lines)

Reflection Symmetry

This transformation provides for mirror symmetric source models the corresponding full mesh and quantity by reflection. In this way a mapping to a full target model is possible, as shown in Figure 9. The user needs to define the Plane's Origin and the Plane's Normal vector of the symmetry plane.
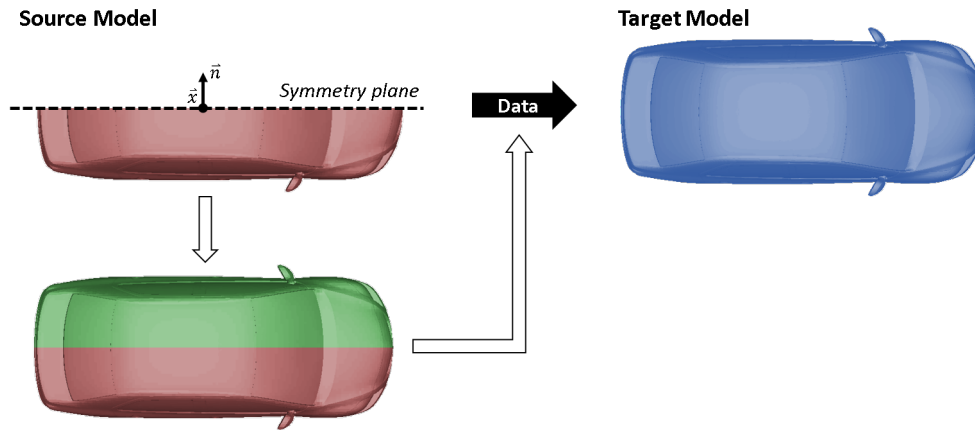
Figure 9: Mapping of data between a mirror symmetric source model and the corresponding full target model.

4.3.2 The "Quantity" Subpanel

In the "Quantity" subpanel (Figure 10), transformations of the quantity can be defined. Following options are available:

None

No quantity transformation is needed.

User defined

Cyclic Symmetry

Enter the cyclic symmetry mode i.e. the Nodal Diameter and the excitation direction (Mode) of the transient or harmonic quantity in order to define the quantity transformation for cyclic symmetric models. For further information refer to ▷ 7.2.2 Cyclic Symmetry of Quantities ◁.

Truncate Transient Data

The data import of transient result data, e.g. pressure and forces, can be truncated using the following options

Filter Transient Id Range

Read transient data beginning and ending at the time step integer IDs given by Start Id and End Id. With the option Step it is possible to read only every n-th time step id.

Filter Time Range

Read transient data beginning at the time value given by Start Time for a defined Period. With the option Step it is possible to read only every n-th time step id.
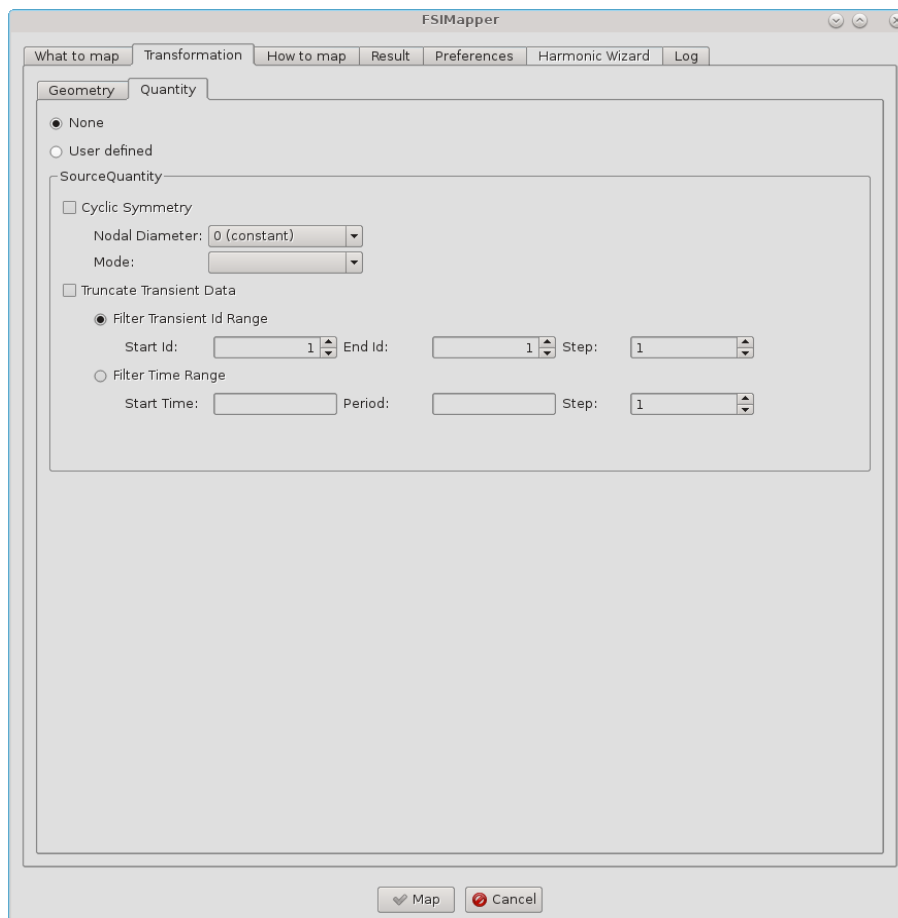
Figure 10: The "Quantity" subpanel of the MpCCI FSIMapper

4.4 The "How to map" Panel

Additional settings concerning the type of algorithm used for the mapping, the necessary parameters, the used orphan fillers or the quantity location can be made in the "How to map" panel (Figure 11).

4.4.1 Mapping Algorithms and Neighborhood Parameters

The MpCCI FSIMapper comes along with three different mapping algorithm approaches that can be used for data interpolation between meshes: the "Shape function", the "Nearest" and the "Weighted Element" algorithm. Fundamental difference between shape function / weighted element and nearest interpolation is the underlying neighborhood relation which is computed. For shape function and weighted element interpolation an element-node relation is required in contrast to nearest interpolation which uses node-node relation. Detailed descriptions of the algorithms can be found in ▷ 7.1 Numerical Methods ◁.

For each type of mapping algorithm there exist different parameters that control the mapping process. After the selection of the algorithm type the relevant parameters are enabled and can be changed below in the "Neighborhood parameters" section (Figure 18).

It is possible to use "Default" parameter selection which should produce good mapping results for many
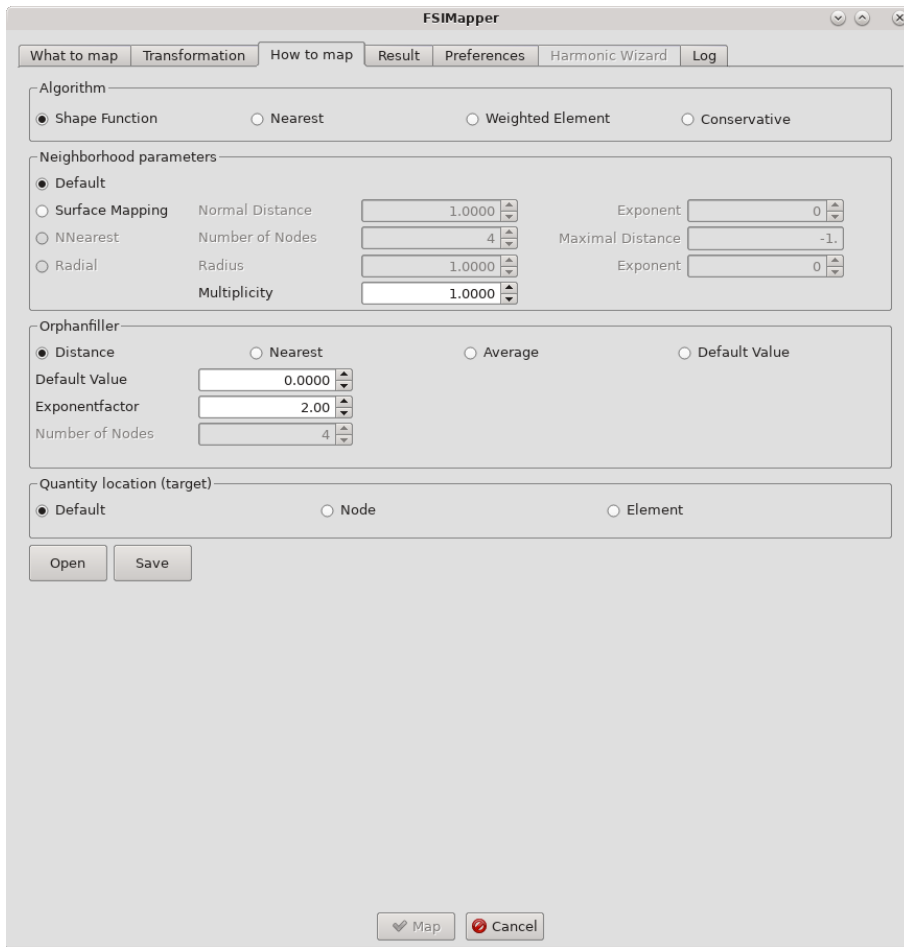
Figure 11: The "How to map" panel of the MpCCI FSIMapper

applications where geometries do not differ too much but they may result in a quite time-consuming neighborhood computation. However, depending on the meshes it might be necessary to adjust the default parameters.

4.4.2 Orphan Filling

Depending on the type of mapping algorithm used, certain elements or nodes of the target geometry may not receive values from the source mesh, for example if the two parts - target and source - do not have the same extension in one direction. These nodes or elements are called "orphans".

The MpCCI FSIMapper provides different methods to assign values to these "orphans". These can be selected in the "Orphanfiller" section on the "How to map" panel. Just like the mapping algorithms the orphan filling methods need some parameters that can be edited after selecting the fill type.

Further information about the different fill methods and parameters can be found in ▷ 7.1 Numerical Methods ◁, where the numerical methods are described.

4.4.3 Quantity Location

Finally – in the last part of the "How to map" panel – the type, i. e. the location of the target quantity can be chosen. By default, the mapped quantity on the target mesh will be located on the same entities (either element or node) as the source quantity.

Additionally the possibility exists to force the target quantity values to be located on either elements or nodes – independent of the source quantity location.

4.4.4 Saving Mapping Configurations

At the bottom of the "How to map" panel the button "Save" provides the capability to store the current configuration – the selected algorithms and parameters – in a configuration file. This offers the possibility of interactive adjustment and optimization of neighborhood parameters for future use in batch mode production runs.
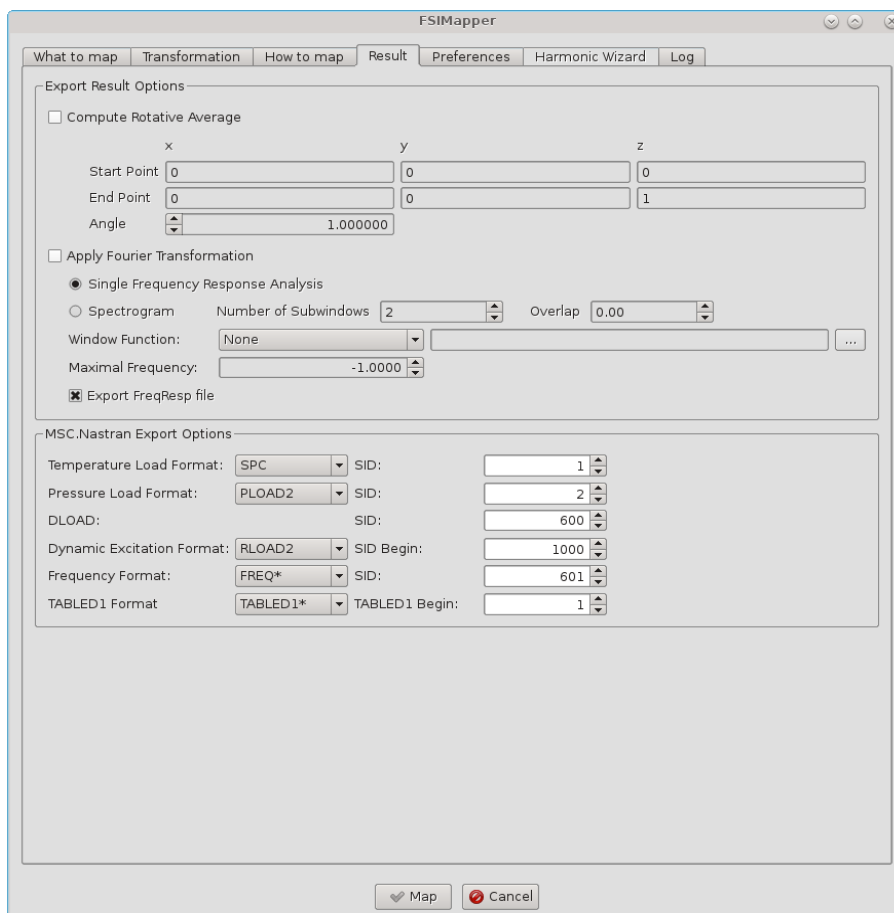
4.5 The "Result" Panel

Figure 12: The "Result" panel of the MpCCI FSIMapper

4.5.1 **Average over Rotation Axis**

The first part of the "Result" panel is only relevant if the CFD simulation uses a so-called "Frozen Rotor" model. In this case the rotation of a geometry part is not modeled with a moving geometry but with the help of additional source terms. This makes the simulation of rotating parts a lot simpler.

The FSIMappper offers the possibility to average these simulation results of a frozen rotor computation by mapping the values on the rotating CSM mesh. Select "Compute Rotative Average" when required.

After ticking the box "Export Result Options" the rotation axis of the target mesh and rotation angle in angular degree need to be specified. The mapping process then is done for a 360° rotation of the target model split up into N substeps where each step is defined by a rotation by the user defined rotation angle around the axis defined through axis start and end point.

4.5.2 **Apply Fourier Transformation**

MpCCI FSIMapper allows to perform Fourier transformations of transient quantities (currently pressure, force and force density supported). The resulting complex quantities can be used in frequency response or spectrogram analyses.

To export loads for this type of analyses select Apply Fourier Transformation when having selected the "Transient" panel in the "What to map" panel.

The following variants are available:

Single Frequency Response Analysis
> The transient data is mapped to the target model and for each node/element the quantity is reformulated by the Fourier transformation as frequency dependent complex load.

Spectrogram
> The time period is divided into several, equally long, potentially overlapping subwindows. In each subwindow a Fourier transformation is performed in order to create complex frequency dependent loads changing over time. Define the Number of Subwindows and the Overlap (between 0 and 0.99). For a signal of a total length L, a number of subwindows m and an overlap of p, the subwindows have a size of $L_S = \lfloor L/(m - (m-1) \cdot p) \rfloor$ time steps. The overlap consists of $\lfloor L_S \cdot p \rfloor$ time steps.
>
> Windowing is recommended here.

The created complex data is only exported up to the given Maximal Frequency, where the default -1 means no export truncation.

Windowing is available through the selection of a pre- or user defined Window Function, see ▷ 7.2.1 Fourier Transformation and Windowing ◁.

(!) The time steps have to be constant and at least two time steps need to be present in the transient source model for the use of this option.

(!) If the time steps are only equidistant within a certain tolerance, MpCCI FSIMapper will take the mean time step calculated by the difference of the last and the first time step divided by the number of time steps minus 1. A warning is displayed in the Log panel.

For further information about the Fourier transformation refer to ▷ 7.2.1 Fourier Transformation and Windowing ◁.

For visualization of the complex data a .ccvx file is written, where for each frequency content of the time signal, the complex quantity (here $\hat{q} = x + \mathrm{i}y$) is inverse Fourier transformed into n time steps by

$$q_j = A \cdot \cos(2\pi \cdot j/n + \phi), \quad j = 1, \ldots, n$$

with the amplitude $A = \sqrt{x^2 + y^2}$ and the phase shift $\phi = \mathrm{atan2}(y, x)$. For vector quantities this is done

component-wise.

The ccvx-file can be read into MpCCI Visualizer by File →Open.... By the "Play" button the n generated time steps of the selected frequency ("Scalars" and "Vectors" in "Result Panel") can be visualized.

4.5.3 MSC Nastran Export Options

Mapped quantities can be included as initial or boundary condition in native CSM solver format, i. e. Abaqus input, ANSYS APDL and MSC Nastran Bulk.

For definition of MSC Nastran temperature and pressure loads there exist several card formats and a unique load id (SID) needs to be specified.

Therefore MpCCI FSIMapper supports the export of temperature either in SPC (MSC Nastran single point constraint) or TEMP format, pressure can be exported as PLOAD, PLOAD2 or PLOAD4. In "Result" panel (Figure 12) both temperature and pressure format can easily be selected in a combo box and unique SID can be assigned.

4.6 The "Preferences" Panel
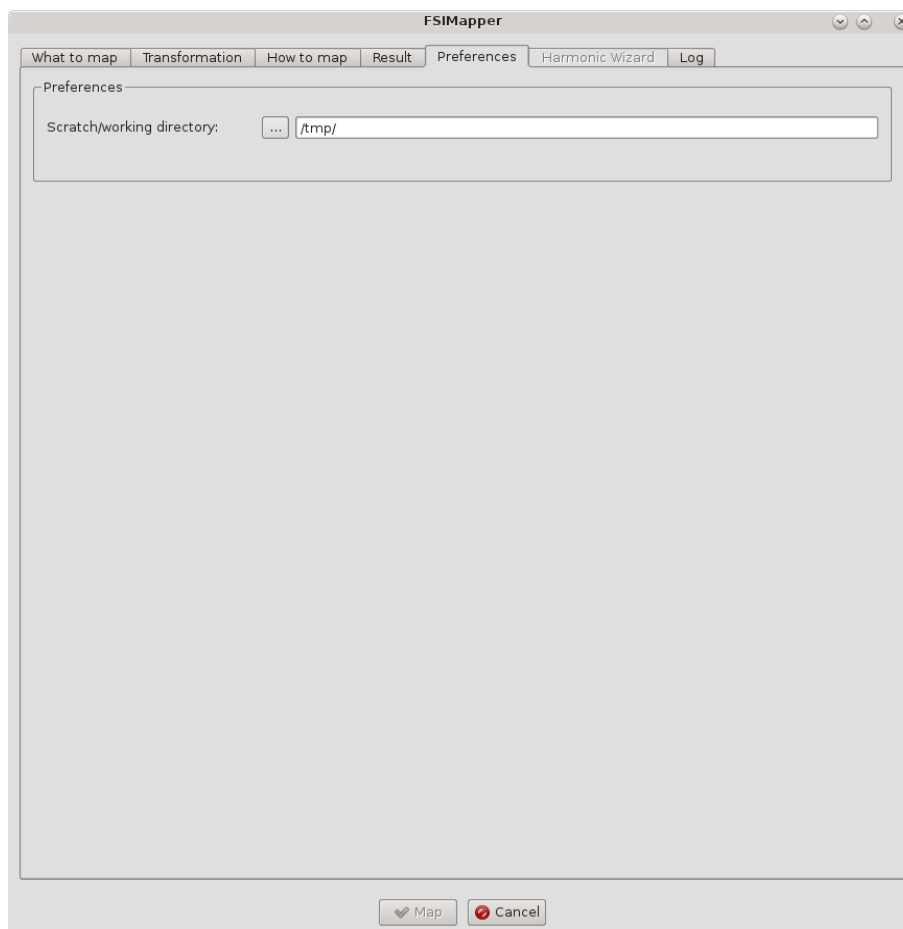
Figure 13: The Preferences panel of the MpCCI FSIMapper

The "Preferences" panel allows just two minor changes: first the user can change the working directory of the MpCCI FSIMapper. The files that are written during the mapping – .vtk files for visualization and include files for a CSM simulation – can be found in the working directory.

Additionally, a box can be ticked if all intermediate files shall be deleted after the mapping is finished.

4.7 The "Harmonic Wizard" Panel
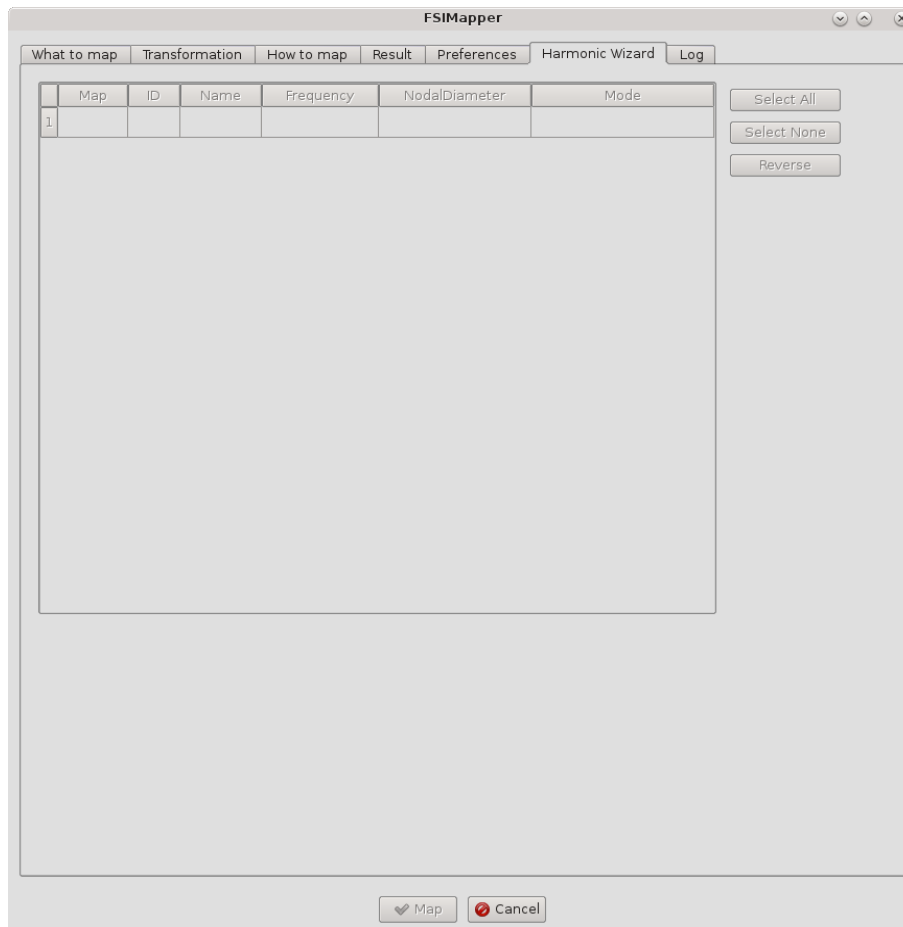
Figure 14: The "Harmonic Wizard" panel of the MpCCI FSIMapper

The "Harmonic Wizard" is designed to map a set of harmonic turbomachinery CFD results at once. It prepares all the necessary information for a mapping between periodic models.

By default, the "Harmonic Wizard" panel is disabled. It is only available for the mapping of harmonic pressure fields simulated by Fidelity Fine Turbo, where additionally the Fidelity Fine Turbo .run file is given by the user. It can be activated by selecting the check box labeled "Use Harmonic Wizard" at the "What to map/Harmonic" panel.

The "Harmonic Wizard" panel lists all harmonic pressure quantities available in the Fidelity Fine Turbo .run file for the selected source parts (which have to belong to one single row). Also the information about the excitation shape (forward/backward nodal diameter) is given for each harmonic, which is used for the mapping between periodic models.

The mapping process (started by pressing the "Map" button) will map the selected harmonics one by one. The mapping results are saved in folders with the naming convention "Row-*RowID*_H-*HarmonicID*".

4.8 The "Log" Panel

Figure 15: The "Log" panel of the MpCCI FSIMapper

The MpCCI FSIMapper has a fixed workflow which can be viewed in the "Log" panel:

1. Evaluation of configuration file content

2. Reading source mesh, quantities and unit conversions

3. Reading target mesh

4. Model positioning if requested

5. Prepare mapping setup

6. Mapping of quantities

7. Export results to files

5 Codes and Formats Information

5.1 Abaqus

5.1.1 Limitations

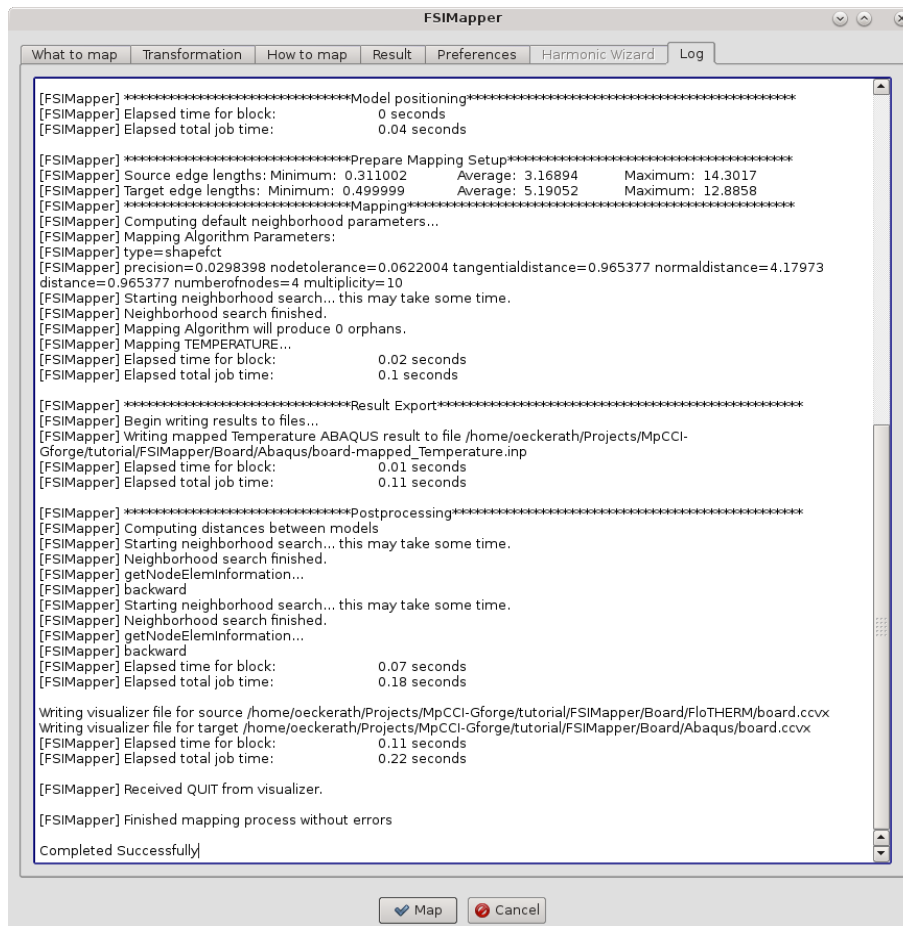**Writing**   For an Abaqus CSM simulation the five quantities "Temperature", "Heat Transfer Coefficient", "Heat Flux", "Pressure" and "Force" can be exported on surface definitions, "Temperature" can also be exported as solid initial condition.

Reading   MpCCI FSIMapper supports reading of the Abaqus ASCII input format for static simulation results. Following quantities (with keywords) are supported: nodal temperature (`*TEMPERATURE`), thermal surface film coefficients (`*SFILM`) and nodal force (`*CLOAD`).

5.1.2 Model Preparation

The Abaqus model can be prepared with Abaqus/CAE or as an input file. Please consider the following advice:

- The model can be defined in either SI or mm-t-s unit systems to be specified in the 'What to map' panel of the MpCCI FSIMapper GUI.

- The Abaqus model must contain a definition of coupling components.
    - This can be either an element-based surface for surface coupling:
        Abaqus/CAE: Create a surface using the Surfaces tool. See also "13.7.6 Using sets and surfaces in the Assembly module" in the Abaqus/CAE User's Manual. You can also use surfaces defined in the Part module.
        Input file: A surface is created with `*SURFACE, NAME=<surface name>, TYPE=ELEMENT`, see section "2.3.3 Defining element-based surfaces" of the Abaqus Analysis User's Manual.

        The surface which serves as coupling component can be selected (marked with 'S' for surface) in the 'What to map' panel of the MpCCI FSIMapper GUI.
    - This can be either an element set for volumetric temperature mapping:
        Abaqus/CAE: Create an element set using the Sets module. See also "13.7.6 Using sets and surfaces in the Assembly module" in the Abaqus/CAE User's Manual.
        Input file: An element set is created with `*ELSET, NAME=<elset name>`, see section "2.2.1 Element definition" of the Abaqus Analysis User's Manual.

        The elset which serves as coupling component can be selected (marked with 'E' for elset) in the 'What to map' panel of the MpCCI FSIMapper GUI.

5.1.3 Include Boundary Conditions

To use the mapped quantity values as boundary conditions in an Abaqus CSM simulation a separate Abaqus input file is written for each boundary condition. This file, which is saved in the model directory, can be included in the original Abaqus input deck using the simple Abaqus include option in the input deck.

Include a mapped temperature by:
`*INCLUDE, INPUT="abaqusFile-mapped_FilmTemp.inp"`
Include a mapped temperature and heat transfer coefficient by:
`*INCLUDE, INPUT="abaqusFile-mapped_FilmTempHTC.inp"`
Include a mapped heat flux by:

```
*INCLUDE, INPUT="abaqusFile-mapped_HeatFlux.inp"
```
Include a mapped pressure field by:
```
*INCLUDE, INPUT="abaqusFile-mapped_Pressure.inc"
```
Include a mapped transient pressure field by:
```
*INCLUDE, INPUT="abaqusFile-mapped_TransientPressure.inc"
```
Include a mapped transient pressure field which was Fourier transformed by:
```
*INCLUDE, INPUT="abaqusFile-mapped_FreqRespPressure.inc"
```
Include a mapped harmonic pressure field by:
```
*INCLUDE, INPUT="abaqusFile-mapped_HarmonicPressure_RE.inc"          and
*INCLUDE, INPUT="abaqusFile-mapped_HarmonicPressure_IM.inc"
```
Include a mapped force field by:
```
*INCLUDE, INPUT="abaqusFile-mapped_Force.inc"
```
Include a mapped transient force field by:
```
*INCLUDE, INPUT="abaqusFile-mapped_TransientForce.inc"
```
Include a mapped transient force field which was Fourier transformed by:
```
*INCLUDE, INPUT="abaqusFile-mapped_FreqRespForce.inc"
```
Include a mapped transient temperature using amplitude definition:
```
*INCLUDE, INPUT="abaqusFile-mapped_Amp.inc"
*INCLUDE, INPUT="abaqusFile-mapped_Temp_Amp.inc"
```

To make sure that the values can be used for an Abaqus analysis, the include has to be placed between the *STEP and *END STEP keywords. Furthermore, for all thermal quantities the step has to be defined as a heat transfer step, e.g. under *COUPLED TEMPERATURE-DISPLACEMENT for a thermal stress analysis. Please refer to the Abaqus keyword manual for a detailed use.

(!) Definition of a transient temperature using amplitude definition normally requires definition of initial reference temperature, e.g. *Initial Conditions, type=TEMPERATURE. Here also a statically mapped temperature can be used.

5.1.4 Elements

MpCCI FSIMapper	stress/displ.	heat transf./mass diff.	heat transf. conv./diff.	acoustic
3D_VOL_HEX8	C3D8	DC3D8	DCC3D8	AC3D8
3D_VOL_HEX20	C3D20	DC3D20	DCC3D20	AC3D20
3D_VOL_HEX27	C3D27	DC3D27	DCC3D27	AC3D27
3D_VOL_TET4	C3D4	DC3D4	DCC3D4	AC3D4
3D_VOL_TET10	C3D10	DC3D10	DCC3D10	AC3D10
3D_VOL_WEDGE6	C3D6	DC3D6	DCC3D6	AC3D6
3D_VOL_WEDGE15	C3D15	DC3D15	DCC3D15	AC3D15
2D_VOL_TRIA3	CPS3			
2D_VOL_QUAD4	CPS4			
2D_VOL_TRIA6	CPS6			
2D_VOL_QUAD8	CPS8			

Table 1: Supported Abaqus solid (continuum) elements. Type variants like reduced integration or coupled temperature-displacement are handled as base type.

MpCCI FSIMapper	stress/displ.	heat transfer
3D_SHELL_TRIA3	S3	DS3
3D_SHELL_TRIA6	S6	DS6
3D_SHELL_QUAD4	S4	DS4
3D_SHELL_QUAD8	S8	DS8
3D_SHELL_QUAD9	S9	

Table 2: Supported Abaqus structural elements. Type variants like reduced integration or coupled temperature-displacement are handled as base type.

5.1.5 Supported Quantities

Quantity	Dim.	Unit (see Table 3)	Location	Abaqus Keyword
TEMPERATURE	Scalar	θ	*Elset	*TEMPERATURE
FILMTEMPERATURE	Scalar	θ	*Surface	*SFILM
HTC	Scalar	$JT^{-1}L^{-2}\theta^{-1}$	*Surface	*SFILM
WALLHEATFLUX	Scalar	$JL^{-2}T^{-1}$	*Surface	*DFLUX
PRESSURE	Scalar	$ML^{-1}T^{-2}$	*Surface	*DLOAD
FORCE	Vector	MLT^{-2}	*Surface	*CLOAD

Unit	Length	Mass	Temperature	Time	Energy
Symbol	L	M	θ	T	J

Table 3: Abaqus unit symbols.

5.2 ANSYS

5.2.1 Requirements

To use MpCCI FSIMapper for file based mapping to ANSYS you need the following:

- Ordinary ANSYS installation.
- It is required that ANSYS can run in standalone correctly and can get a license without any issue.

5.2.2 Model Preparation

There are some issues which you should consider while creating an ANSYS model for a co-simulation:

Supported element types. Not all ANSYS element types are supported, the supported types are given in Table 4.

Dummy surface elements for surface coupling. For surface coupling, additional surface elements must be used for quantity exchange. For instance SHELL63 elements for a fluid structure interaction or SHELL57 elements for thermal coupling can be used as dummy elements to receive forces from the partner code. The dummy elements must have the corresponding quantities you want to receive or send. Only these elements take part in the coupling process and must either be deselected when the solution is performed or defined so weak that they do not influence the solution in case of fluid structure interaction. They can be deselected before solution is executed if only nodal quantities

ANSYS Element Types	**MpCCI FSIMapper**
PLANE13, PLANE25, PLANE42, PLANE55, PLANE67, PLANE181, PLANE182, SHELL28, SHELL41, SHELL43, SHELL57, SHELL63, SHELL131 SHELL143, SHELL157, HYPER56, VISCO106, CPT212, SURF252,	3D_SHELL_TRIA3, 3D_SHELL_QUAD4
PLANE2, PLANE35	3D_SHELL_TRIA3, 3D_SHELL_TRIA6
SHELL91, SHELL93, SHELL99, SHELL132, SHELL150, SHELL281 PLANE53, PLANE145,PLANE223, PLANE77, PLANE78, PLANE82, PLANE83, PLANE121, PLANE183, PLANE146, PLANE230 SURF152, SURF154 CPT213, HYPER74, VISCO88, VISCO108,	3D_SHELL_TRIA6, 3D_SHELL_QUAD4, 3D_SHELL_QUAD8
SOLID5, SOLID45, SOLID46, SOLID62, SOLID64, SOLID65, SOLID69, SOLID70, SOLID96, SOLID97, SOLID164, SOLID285, SOLID185, HYPER58, HYPER86, VISCO107, SOLSH190, CPT215	3D_VOL_TET4, 3D_VOL_WEDGE6, 3D_VOL_HEX8, 3D_VOL_PYRAM5
SOLID87, SOLID92, SOLID98, SOLID123, SOLID127, SOLID148, SOLID168, SOLID186,SOLID187, SOLID227, SOLID232,SOLID237, CPT217	3D_VOL_TET4, 3D_VOL_TET10
SOLID90, SOLID95, SOLID122, SOLID117, SOLID122, SOLID128, SOLID147, SOLID186, SOLID191, SOLID226, SOLID231, SOLID236, VISCO89, CPT216	3D_VOL_TET4, 3D_VOL_TET10, 3D_VOL_WEDGE6, 3D_VOL_WEDGE15, 3D_VOL_HEX8, 3D_VOL_HEX20, 3D_VOL_PYRAM5, 3D_VOL_PYRAM13
MESH200	MPCCI_ETYP_LINE2, MPCCI_ETYP_TRIA3, MPCCI_ETYP_QUAD4, MPCCI_ETYP_QUAD8

Table 4: Supported ANSYS element types.

are sent or received, because the data transfer will then put the values to the nodes and the nodes of dummy elements are shared by the "real" solid model elements. In case of thermal coupling the dummy elements can not be deselected because the quantities wall heat transfer coefficient, wall temperature and wall heat flux are only supported as element quantities. Therefore the additional layer of shell elements (SHELL57) have to be a part of the solution. To reduce the influence on the solution you should give the shell elements the same material properties as the adjacent solid elements and a small thickness. Such dummy elements are shown in Figure 16

Put elements for coupling in a component. The elements of the coupling region must be grouped into one or more element components using the `cm` command.

Predefined loads on dummy surface elements. If element based quantities HTC, TEMPERATURE, WALLHEATFLUX or PRESSURE are received by ANSYS using dummy surface elements, you have to predefine surface loads on these elements in order to enable the MpCCI ANSYS adapter to select the correct element sides to store the received values. Therefore select the coupled element set and the attached nodes and define a dummy load, depending on the received quantities:

- HTC and TEMPERATURE: `~SF, ALL, CONV, 1, 300`
- WALLHEATFLUX: `~SF, ALL, HFLUX, 1`
- PRESSURE: `~SF, ALL, PRES, 1`

Possible quantities depend on degrees of freedom. Normally the degree of freedom of the elements involved in the coupling process determines which quantities can be transferred. It is laborious to find out if all degrees of freedom are actually supported by the ANSYS API. As this API is used for the MpCCI ANSYS–adapter, it is not guaranteed that all theoretically supported degrees of freedom are valid.

Not carefully tested. Only few of the element type mappings are already validated with certain quantities. The compatibility index in Table 5 shows the validated element-quantity pairs. It is constantly added. Please contact us if you have problems with other combinations or if you need additional capabilities.
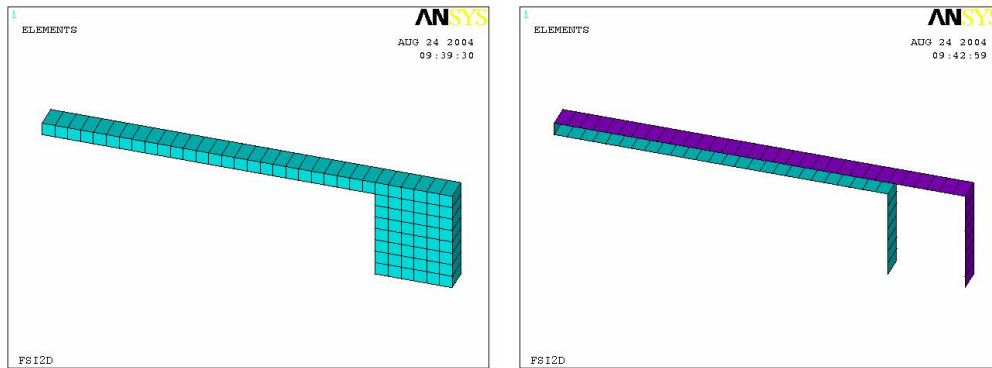

Figure 16: ANSYS volume model and additional dummy SHELL63 elements for surface coupling

Quantity / *Element*	TEMPERA-TURE	HTC	WALL-HEATFLUX	PRESSURE
SOLID5				+
SOLID45				+
SHELL57	+	+	+	
SHELL63				+
SHELL93				+
SURF151	+	+	+	

Table 5: Quantities supported by MpCCI FSIMapper for different ANSYS elements.

5.2.3 Supported Quantities

Quantity	Dim.	Unit	Location	ANSYS Keyword
TEMPERATURE	Scalar	θ	Solid/Shell	BF,,,TEMP
HTC	Scalar	$JT^{-1}L^{-2}\theta^{-1}$	Shell	SFE,,,CONV
WALLHEATFLUX	Scalar	$JL^{-2}T^{-1}$	Shell	SFE,,,HFLUX
PRESSURE	Scalar	$ML^{-1}T^{-2}$	Shell	SFE,,,PRES

5.2.4 ANSYS Scanner and Converter

ANSYS provides two native solver file formats. The .cmd format is ASCII based whereas the more common data base .db format is binary. The MpCCI FSIMapper makes use of the MpCCI code adapter to convert

both .cmd and .db format into an own intermediate format.

When selecting code ANSYS as target model in "What to map" panel of the MpCCI FSIMapper GUI and specifying either a .cmd or .db file, the model automatically gets converted into intermediate format. This is done by executing the command

`> fsimapper convert ANSYS modelFile.db` (see ▷ 3 MpCCI FSIMapper Command ◁ for details)

creating a new .ml file in same system folder where original model file is located. This result file is automatically selected as target model in the "What to map" panel.
Then the file scanner for intermediate format .ml is launched by

`> fsimapper scan ANSYS modelFile.ml`

listing available components that have been converted for a mapping.

5.3 ANSYS CFX

MpCCI FSIMapper supports reading of CFX results exported in "CFD-Post Generic Export" format.

5.3.1 Supported Quantities

Quantity	Dim.	Unit	Location
FILMTEMPERATURE	Scalar	K	Face
HTC	Scalar	$W/(m^2\ K)$	Face
PRESSURE	Scalar	N/m^2	Face

5.4 ANSYS Maxwell

MpCCI FSIMapper supports reading of ANSYS Maxwell export to Universal file format .unv. Universal file node blocks (781 and 2411), element blocks (780 and 2412) and quantity block 2414 with node and element location are supported.

5.4.1 Elements

3D_SURF_TRIA3
3D_SURF_TRIA6
3D_SURF_QUAD4
3D_VOL_TET4
3D_VOL_TET10
3D_VOL_HEX8
3D_VOL_HEX20
3D_VOL_WEDGE6

Table 6: Supported structural elements in Universal file.

5.4.2 Supported Quantities

Quantity	Dim.	Default Unit	Location	Comments
FORCE	Vector	N	wall	Surface Force

5.5 EnSight Gold

MpCCI FSIMapper supports reading and writing of binary EnSight Gold .case and .encas files for static, transient and harmonic simulation results. Only scalar quantities are supported.

EnSight Gold only knows 3–dimensional meshes. If a mapping is performed between the surfaces of 2-dimensional models, "EnSight Case (2D)" must be selected; it converts internally the 3-dimensional model to a 2-dimensional model by removing the third degree of freedom (z).

5.5.1 Supported Quantities

Due to the open structure of the EnSight Gold Case format, MpCCI FSIMapper supports reading of all saved scalar and vector quantities.

5.6 Fidelity Fine Turbo

MpCCI FSIMapper supports reading of Fidelity Fine Turbo native .cgns files for static and harmonic simulation results. The corresponding .run file (model setup) can be imported in order to insert the surface and volume names originally defined in Fidelity Fine Turbo (instead of the internally defined names in the .cgns file).

(!) Mapping is possible only if Fidelity Fine Turbo was executed using the expert parameter ICGP3D is set to 1

(!) If the harmonic wizard is to be used, the import of the .run file is necessary

5.6.1 Supported Quantities

Quantity	Dim.	Default Unit	Location	comments
TEMPERATURE	Scalar	K	wall	
HEATTRANSFERCOEF	Scalar	$W/(m^2\ K)$	wall	
HEATFLUX	Scalar	W/m^2	wall	
PRESSURE	Scalar	N/m^2	wall	
REP_x IMP_x	Scalar	N/m^2	volume	complex harmonic pressures (NLH method)

Fidelity Fine Turbo only saves the results of the Nonlinear Harmonic method in the fluid volume and not on the wall.

5.7 FloEFD

MpCCI FSIMapper supports reading of FloEFD export format .efdfea available in version 13 for static simulation results.

5.7.1 Supported Quantities

Quantity	Dim.	Default Unit	Location
TEMPERATURE	Scalar	K	solid/wall
HTC	Scalar	$W/(m^2\ K)$	wall
SURFACE_HEAT_FLUX	Scalar	W/m^2	wall
PRESSURE	Scalar	N/m^2	wall
RELATIVE_PRESSURE	Scalar	N/m^2	wall

5.8 FloTHERM

MpCCI FSIMapper supports reading of FloTHERM export format .flofea available in version 10 for static simulation results.

5.8.1 Supported Quantities

Quantity	Dim.	Default Unit	Location
TEMPERATURE	Scalar	C	solid

5.8.2 Combining Static Result Files

MpCCI FSIMapper supports the combination of several static FloTHERM result files of the same geometry by creating a tabular master input file. To assign the individual time step information per *.flofea result file, a comma separated two-column format is used. The master *.csv file has to be located in the same folder to address the individual time step location. The following example shows the master file structure for a mapping setup with four different time step results.

```
#comment line
#time value, file name
 0.0, result_at_0.0.flofea
 5.0, result_at_5.0.flofea
10.0, result_at_10.0.flofea
20.0, result_at_20.0.flofea
```

5.9 FLUENT

MpCCI FSIMapper supports reading of FLUENT native .cas (geometry) and .dat (solution result) files for static simulation results. In general .cas and .dat files between FLUENT version 6.3.26 and version 20.1.x are supported and more recent versions should be compatible if native file format does not change. FLUENT CFF files (.h5) are supported from FLUENT 20.0.

(!) Only zone types "wall" and "fluid" can be coupled with MpCCI FSIMapper.

5.9.1 Supported Quantities

Quantity	Dim.	Default Unit	Location	Comments
TEMPERATURE	Scalar	K	fluid	
FILMTEMPERATURE	Scalar	K	wall	
HTC (computed)	Scalar	$W/(m^2\ K)$	wall	$\alpha = q/(T_w - T_c)$
wallfuncHTC	Scalar	$W/(m^2\ K)$	wall	stored in UDM0
WALLHEATFLUX	Scalar	W/m^2	wall	
STATICPRESSURE	Scalar	N/m^2	wall	
ABSPRESSURE	Scalar	N/m^2	wall	STATICPRESSURE + operating pressure
OVERPRESSURE	Scalar	N/m^2	wall	STATICPRESSURE + operating pressure - reference pressure
WALLFORCE	VECTOR	N	wall	ABSPRESSURE + SHEARFORCE
RELATIVEWALLFORCE_STATIC	VECTOR	N	wall	STATICPRESSURE + SHEARFORCE
RELATIVEWALLFORCE_OVER	VECTOR	N	wall	OVERPRESSURE + SHEARFORCE

5.9.2 Exporting wallfuncHTC to UDM0 in Data File

For a surface heat transfer analysis "FILMTEMPERATURE" as well as the wall heat transfer coefficient need to be mapped onto a CSM model. The latter, defined as

$$\alpha = \frac{Q}{A\Delta T}$$

where

- Q = heat flow, J/s = W

- α = heat transfer coefficient, $W/(m^2 K)$

- A = heat transfer surface area, m^2

- ΔT = difference in temperature between the solid surface and surrounding fluid area, K.

It can be displayed in the FLUENT GUI Display/Graphics and Animations/Contours/Set Up/Wall Fluxes.. /Wall Func. Heat Tran. Coef.. By default the wall heat transfer coefficient is not stored to a FLUENT data (.dat) file but can be added by following the steps:

1. Define/User-Defined/Memory:
   Set number to 1

2. Define/Custom-Field-Function:
   Select "wallfuncHTC" below "Wall Fluxes".
   Accept by 'Define'.

3. Solve/Initialization/Patch:
   Activate "Use Field Function" and select defined custom field function from previous step.
   Select all fluids and solids as well as "Variable User Memory 0" at bottom left side.
   Select "Patch".

4. Case/Data:
   Save HTC to new .dat file.

5.10 JMAG

MpCCI FSIMapper supports reading of JMAG export format MSC Nastran .bdf available from the Multi-Purpose File Export Tool. This tool requires the JMAG results file "*.jplot" for the export process.

5.10.1 Limitations

- The Multi-Purpose File Export Tool is only available from JMAG Designer on Windows platform. The access to the tool is provided under the menu Tool→Multi-Purpose File Export Tool....

- Please check the restrictions on the MSC Nastran format for reading the data. MpCCI FSIMapper supports force and heat source definition from MSC Nastran format (▷5.13 MSC Nastran ◁).

- Output type is a Static Analysis from JMAG.

### 5.10.2 Exporting JMAG Data to MSC Nastran Format With Multi-Purpose Export

1. Open the Multi-Purpose File Export Tool dialog.

2. Select Conversion Type to Nastran/CSV/Universal File to Convert Files From Jplot.

3. Select Output File Type to MSC Nastran File(Free Field)

4. Provide the output file name in the Output File field.

5. Click the Add... button from the Input Result files.
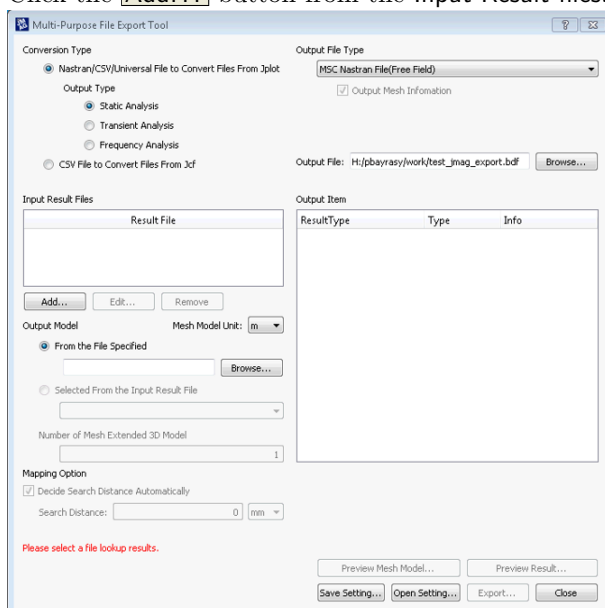
6. The next dialog to select the JMAG result file is opening. Pick up the "jplot" file with the Browse.
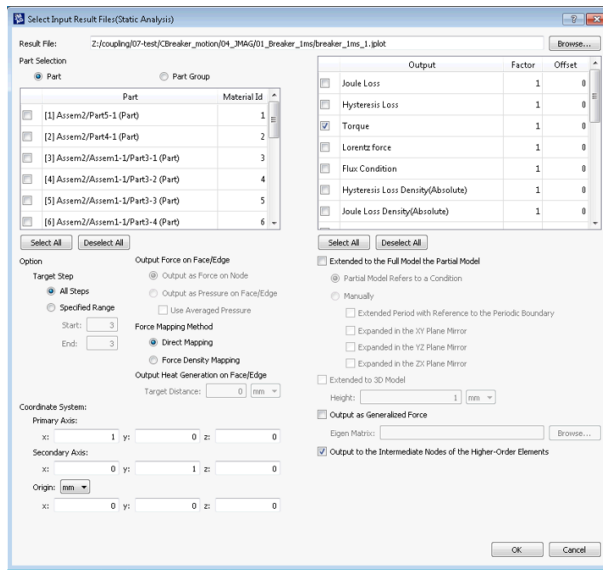
7. Check the parts you want to export the values from in the optionPart selection.

8. Select the output value you want to export from the model.

9. Close this dialog on pressing $\boxed{\text{OK}}$.

10. Back to the main dialog, select under Output Model the Selected From the Input Result File in the droplist.

11. Finish by clicking on $\boxed{\text{Export...}}$ button.

12. The exported file is used as source file in MpCCI FSIMapper. Select the MSC Nastran code for this purpose.
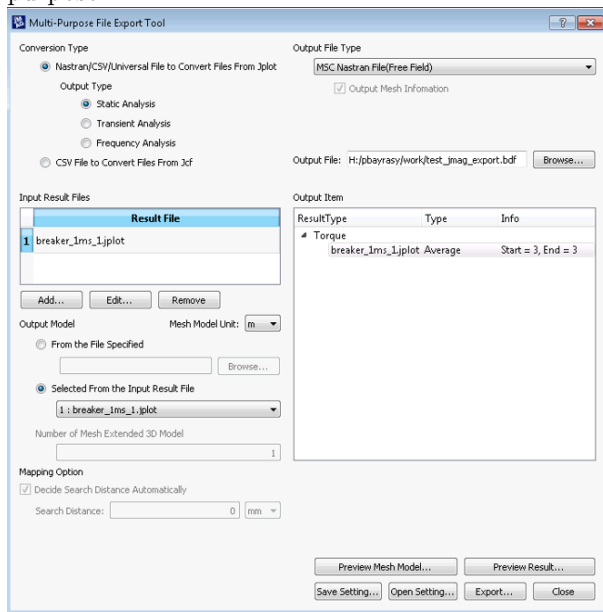
5.11 LS-DYNA

MpCCI FSIMapper supports code LS-DYNA as mapping target using it's ASCII keyword data format.

5.11.1 Limitations

The structural code LS-DYNA is only available for "Static" or "Transient" applications. On "Harmonic" source panel, the code will not be available in the target code dropdown menu. The quantity TEMPERATURE can be transferred to either solid or shell elements, PRESSURE and FORCES can only be used on shell elements. If PRESSURE or FORCES shall be transferred onto a solid model, it is required to define additional shell elements on solid outer surface.

5.11.2 Elements

LS-DYNA	MpCCI FSIMapper
3 node shell elements	3D_SHELL_TRIA3
4 node shell elements	3D_SHELL_QUAD4
4 node solid elements	3D_VOL_TETRA4
6 node solid elements	3D_VOL_WEDGE6
8 node solid elements	3D_VOL_HEXA8

Table 7: Supported LS-DYNA elements.

5.11.3 Supported Quantities

Quantity	Dim.	Default Unit	Element	LS-DYNA Keyword
TEMPERATURE (static, transient)	Scalar	θ	Shell Volume	*INITIAL_TEMPERATURE_NODE *LOAD_THERMAL_VARIABLE_NODE *DEFINE_CURVE
PRESSURE (static, transient)	Scalar	$ML^{-1}T^{-2}$	Shell	*LOAD_SHELL_ELEMENT *DEFINE_CURVE
PRESSURE (frequency)	Scalar	$ML^{-1}T^{-2}$	Shell	*DATABASE_FREQUENCY_BINARY_D3SSD *FREQUENCY_DOMAIN_SSD *DEFINE_CURVE
FORCE (static, transient)	Vector	MLT^{-2}	Shell	*LOAD_NODE_POINT *DEFINE_CURVE
FORCE (frequency)	Vector	MLT^{-2}	Shell	*DATABASE_FREQUENCY_BINARY_D3SSD *FREQUENCY_DOMAIN_SSD *DEFINE_CURVE

Table 8: Mapping variables overview with used LS-Dyna keywords.

5.11.4 Include Boundary Conditions

To use the mapped quantity values as boundary conditions in an LS-DYNA CSM simulation a separate LS-DYNA input file is written for each boundary condition. This file, which is saved in the model directory, can be included in the original LS-DYNA input deck using the simple LS-DYNA include option in the input deck.

To include a mapped result file use the statement

TEMPERATURE static	*include lsdynaFile-mapped_Temperature.inc
TEMPERATURE transient	*include lsdynaFile-mapped_Transient_Temperature.inc
PRESSURE static	*include lsdynaFile-mapped_Pressure.inc
PRESSURE transient	*include lsdynaFile-mapped_Transient_Pressure.inc
PRESSURE frequency	*include lsdynaFile-mapped_FreqRespPressure.inc
FORCE static	*include lsdynaFile-mapped_Forces.inc
FORCE transient	*include lsdynaFile-mapped_Transient_Forces.inc
FORCE frequency	*include lsdynaFile-mapped_FreqRespForce.inc

5.12 MagNet

MpCCI FSIMapper supports reading of MagNet export format .vtk available from version 7.5.1 for static and transient simulation results. Multiple components are written to distinct .vtk files which can be inputted to MpCCI FSIMapper by multi selection. Mapping of static and transient data is possible.

5.12.1 Limitations

Currently only stationary components can be considered for mapping in both static and transient analyses.

5.12.2 Elements

MpCCI FSIMapper	VTK Element Type
3D_SURF_TRIA3	5
3D_SURF_TRIA6	22
3D_SURF_QUAD4	9
3D_SURF_QUAD8	23
2D_SURF_LINE2	3
2D_SURF_LINE3	21

Table 9: Supported MagNet (.vtk) structural elements.

5.12.3 Supported Quantities

Quantity	Dim.	Default Unit	Location	Comments
SFD	Vector	N/m^2	wall	Surface Force Density

5.13 MSC Nastran

5.13.1 Limitations

Writing   As boundary conditions for a CSM simulation with Nastran temperature, pressure or force values can be exported. For temperature values the keywords TEMP or SPC are used and the quantity values are mapped to and written on nodes. The keywords PLOAD, PLOAD2 and PLOAD4 are used for the export of element-based pressure values (see also ▷ 4.5.3 MSC Nastran Export Options ◁). Forces from a static mapping are exported using the keywords FORCE, a transient mapping uses the keywords TSTEP, TLOAD1,

`DAREA` and `TABLED1`. When a Fourier transformation has been performed on transient data, then the keywords `FREQ`, `RLOAD1`, `RLOAD2`,`DAREA` and `TABLED1` are used.

(!) The Nastran mesh has to consist of shell elements for exporting pressure in the `PLOAD*` format.
The Nastran mesh has to consist of shell elements for exporting force.
The Nastran mesh has to consist of shell or solid elements for exporting temperatures in `TEMP` or `SPC` format.

(!) When analysing solid **MSC Nastran** models with surface loads and **MpCCI FSIMapper** can handle only shell elements, create a shell mesh at the coupled surface using the same grids as the solid model. Use this model for the mapping process.

Reading   Moreover, **MpCCI FSIMapper** supports reading of **MSC Nastran** files in bulk data format. For static simulation results the following quantities (with keywords) are supported: temperature (`TEMP` and `TEMP*`), pressure (`PLOAD2` and `PLOAD2*` (single element per line assumed), `PLOAD4` and `PLOAD4*` (assuming a constant value per element)), force (`FORCE` and `FORCE*` (ignoring the coordinate system identification number)) and heat source (`QVOL` and `QVOL*`).
For transient results, only force is supported (defined by the keywords `TLOAD1` (ignoring the delay), `DAREA`, `TABLED1` or their corresponding long formats).

5.13.2 Include Boundary Conditions

To use the mapped quantity values as boundary conditions in an **MSC Nastran** CSM simulation a separate **MSC Nastran** input file is written for each boundary condition. This file, which is saved in the model directory, can be included in the original **MSC Nastran** input deck using the simple **MSC Nastran** include option in the input deck.
The file extension of the exported include files is the same as of the target mesh file. Here, the Bulk Data Entries are shown for .bdf. Include a mapped temperature using the `TEMP` keyword by:
`INCLUDE 'nastranFile-mapped_Temp.bdf'`
Include a mapped temperature using the SPC keyword by:
`INCLUDE 'nastranFile-mapped_Temperature_SPC.bdf'`
Include a mapped pressure field by:
`INCLUDE 'nastranFile-mapped_Pressure.bdf'`
Include a mapped transient pressure field by:
`INCLUDE 'nastranFile-mapped_TransientPressure.bdf'`
Include a mapped transient pressure field which was Fourier transformed by:
`*INCLUDE, INPUT="nastranFile-mapped_FreqRespPressure.bdf"`
Include a mapped harmonic pressure field by:
`*INCLUDE, INPUT="abaqusFile-mapped_HarmonicPressure.bdf"`
Include a mapped force field by:
`INCLUDE 'nastranFile-mapped_Force.bdf'`
Include a mapped transient force field by:
`INCLUDE 'nastranFile-mapped_TransientForce.bdf'`
Include a mapped transient force field which was Fourier transformed by:
`INCLUDE 'nastranFile-mapped_FreqRespForce.bdf'`

To make sure that the values can be used for an **MSC Nastran** analysis, the include has to be placed between the `BEGIN BULK` and `ENDDATA` keywords.
The set identification numbers (**MSC Nastran** `SID` of Bulk Data Entry) defined by **MpCCI FSIMapper** have to be referenced in the Case Control Commands of the **MSC Nastran** analysis deck. Please refer to the **MSC Nastran** keyword reference manual for a detailed use.

5.13.3 Elements

MSC Nastran	MpCCI FSIMapper
CTRIA3	3D_SHELL_TRIA3
CTRIA6	3D_SHELL_TRIA6
CQUAD4	3D_SHELL_QUAD4
CQUAD8	3D_SHELL_QUAD8
CQUAD	3D_SHELL_QUAD9
CTETRA	3D_VOL_TET4/10
CHEXA	3D_VOL_HEX8/20
CPENTA	3D_VOL_WEDGE6

Table 10: Supported MSC Nastran structural elements.

5.13.4 Supported Quantities

Quantity	Dim.	Default Unit	Location	MSC Nastran Keyword
(FILM)TEMPERATURE	Scalar	θ	Shell/Volume	TEMP
				SPC
PRESSURE	Scalar	$ML^{-1}T^{-2}$	Shell	PLOAD
				PLOAD2
				PLOAD4
				RLOAD1
				RLOAD2
FORCE	Vector	MLT^{-2}	Node	FORCE
				TLOAD1
				RLOAD1
				RLOAD2

5.14 VMAP

VMAP is an open and free standard www.vmap.eu.com for the transfer of information between simulations within a process chain, no matter what software is used to carry these out. This provides engineers with a fast solution to pass information from one calculation to another without needing to create one-off interfaces.

VMAP provides a detailed and well-defined open-source standard, accompanied by a software library of input/output tools to read/write the VMAP data files and enable fast development of transfer operations between different software packages. Many software companies and suppliers have included, or are working on including, VMAP into their tools.

The VMAP format is currently available in MpCCI FSIMapper for "static" mapping applications. As VMAP is a neutral file format, the assigment of variables can be done via name selection in the variables drop down menu.

5.15 6SigmaET

6SigmaET is an electronics thermal modeling tool. MpCCI FSIMapper supports reading of both "static" and "transient" simulation results. For "static" use case VTK legacy *.vtk and *.vtm format is supported,

"transient" *.vtm and *.ptu forat is available for reading. Variables TEMPERATURE and PRESSURE can be imported on surface or volume meshes.

6 Batch Usage of the MpCCI FSIMapper

The MpCCI FSIMapper can also be used as a batch tool. All functionalities of the graphical user interface are also available in the batch version.

The basic MpCCI FSIMapper is an executable that needs several arguments for execution: the file names of the source and the target models and a configuration file where all necessary mapping parameters are set (see ▷ 3 MpCCI FSIMapper Command ◁). A detailed description of the configuration file format can be found in ▷ 6.5 Configuration File ◁.

To find the relevant parts of the models for the interface file scanners for FLUENT, Abaqus, ANSYS, FloTHERM, FloEFD and EnSight Gold are part of the MpCCI FSIMapper. These perl modules list all found parts and the assigned ids.

6.1 File Scanners

The MpCCI FSIMapper uses a configuration file to decide which quantities to map between which geometry parts. All geometry parts of a model can be addressed by ids, which can be identified by the file scanners. The user has to enter the relevant ids for the CFD and the FEA mesh into the configuration file.

For a thermal mapping case usually only the ids for the interface between the fluid and solid domain will be entered in the configuration file. The rest of the geometry does not need to be read by the MpCCI FSIMapper. Depending on the type of application the user can decide which geometry parts are to be read by the MpCCI FSIMapper.

To get the ids of the geometry parts, solver files can be scanned by perl modules that are part of the MpCCI FSIMapper package. The only requirement is a running perl installation with its executables in the system PATH variable. All file scanners – for FLUENT, Abaqus, ANSYS, FloTHERM, FloEFD, EnSight Gold, CFX and Fidelity Fine Turbo– can be called via the command line:

```
> fsimapper scan FLUENT  file_name.cas
> fsimapper scan ABAQUS  file_name.inp
> fsimapper scan ANSYS   file_name.ml
> fsimapper scan ENSIGHT file_name.case
> fsimapper scan FLOEFDMAPLIB file_name.efdfea
> fsimapper scan FLOTHERMMAPLIB file_name.flofea
> fsimapper scan FINETURBO file_name.cgns
> fsimapper scan CFXCSV file_name.csv
```

ⓘ MpCCI FSIMapper does not allow scanning of native ANSYS solver formats directly but of converted .ml files generated by "fsimapper convert ANSYS" (▷ 5.2.4 ANSYS Scanner and Converter ◁).

6.1.1 FLUENT

The FLUENT scanner lists all zones that are defined in the .cas file. The names of the zones are listed, as well as the zone type and the zone id. The type of the zones is important to find the parts of the model that form the interface between the fluid and the structural domain.

Usually, the zones of type "wall" are to be considered for the MpCCI FSIMapper. The zone id is used in the configuration file to identify the part of the model from which quantity values shall be mapped to the CSM model.

The format of the output is the following:

```
Zonename           Zonetype              ZoneID
```

6.1.1.1 Example

As an example the output of the FLUENT scanner for a simple case file "exampleFluent.cas" is shown below.

```
> fsimapper scan FLUENT ./exampleFluent.cas
<CPLDATA>
# This file was automatically created for the MpCCI GUI.
#@ Code: "FLUENT"
#@ Scan: "..../MpCCI/codes/FLUENT/Scanner.pm"
#@ File: "exampleFluent.cas"
#@ Path: "..."
#@ Date: "..."
#@ User: "..."
#@ Host: "..."
#
# MpCCI relevant information:
#   Model dimensions  : 3D
#   Coordinate system : Cartesian
#   Solution type     : Steady state
#   Load cases        : 1
#   Unit system       : SI
#   Precision         : Double precision (64 bit)
#
# Additional scanner information:
#   Global variables  : Autogenerated by the scanner
#   Release           : 13.0.0
#   Saved by          : fluent
#   Used UDM          : 0
#   Used UDS          : 0
#   Version           : 3ddp
#
   fluid_1                    fluid                      7
   wall_1                     wall                       64
   interior_1                 interior                   78
   interface_1                interface                  79
</CPLDATA>
```

This FLUENT model consists of four different zones: one of each of the four types "fluid", "wall", "interior" and "interface". The zone with the name wall_1 of type "wall" has the id 64. If we now want to couple wall_1 we have to add the following lines to the configuration file ($\triangleright$ 6.5 Configuration File $\triangleleft$) defining FLUENT as source file format and part 64 to be read:

```
sourceFileType
FLUENT
sourceMeshPartIds
64
```

For the mapping of quantity data usually the zones of type "wall" are of interest, as the interface between

the CFD and FEA domains is normally a wall boundary in the CFD model.

For large FLUENT models and Linux machines it might be helpful to only list the wall zones of the model:
```
> fsimapper scan FLUENT ./exampleFluent.cas | grep wall
```
This gives a clearer overview of the possible interfaces without listing all interior or fluid zones.

6.1.2 MentorGraphics

In cooperation Fraunhofer SCAI and MentorGraphics have developed a neutral file format allowing export of FloEFD and FloTHERM simulation results for mapping to a CSM code.

6.1.3 FloEFD

The scanner for FloEFD export files can be selected by its name FLOEFDMAPLIB in the generic MpCCI FSIMapper scan command, e.g.

```
> fsimapper scan FLOEFDMAPLIB file_name.efdfea
```

The resulting output of the scanner has a simple tabular format with just the keywords MESH, PART or QUANT. MESH assigns the name of the model to be read in. PART specifies 'S' or 'V', if it is a surface or a volume, the part name as well as two integers specifying the total number of nodes and elements of the part. Parts are implicitly numbered by their order of appearance starting at index 1. Keyword QUANT announces a quantity followed by its name, location of the values, the number of values and the file position of the quantity.

6.1.3.1 Example

```
> fsimapper scan FLOEFDMAPLIB ./Plate.efdfea
MESH EFD_MESH
PART   S   Surface-1    302     600
PART   V   Volume-1     390     216
QUANT RELATIVE_PRESSURE ELEM   816     54917
QUANT PRESSURE          ELEM   816     58198
QUANT SURFACE_HEAT_FLUX ELEM   816     61470
QUANT TEMPERATURE       ELEM   816     64751
QUANT HTC               ELEM   816     68026
```

One surface (`Surface-1`) addressed by 'S' and id 1 and one volume (`Volume-1`) addressed by 'V' and with id 2 are defined in this FloEFD model. In addition FloEFD file does contain a list of element based quantities, e.g. RELATIVE_PRESSURE, PRESSURE, SURFACE_HEAT_FLUX, TEMPERATURE and HTC, which can be read from file.

Parts to be mapped can be selected by its ids separated by whitespace in the configuration file. In this example we now want to couple the volume part and therefore have to add the following lines to the configuration file:

```
sourceFileType
FLOEFDMAPLIB
sourceMeshPartIds
2
```

A mapping of the surface `Surface-1` would require to add

```
sourceFileType
FLOEFDMAPLIB
sourceMeshPartIds
1
```

to configuration file.

6.1.4 FloTHERM

The scanner for FloTHERM export files can be selected by its name `FLOTHERMMAPLIB` in the generic MpCCI FSIMapper scan command, e.g.

```
> fsimapper scan FLOTHERMMAPLIB file_name.flofea
```

The resulting output of the scanner has a simple tabular format with just the keywords MESH, PART or QUANT. MESH assigns the name of the model to be read in. PART specifies 'S' or 'V', if it is a surface or a volume, the part name as well as two integers specifying the total number of nodes and elements of the part. Parts are implicitly numbered by their order of appearance starting at index 1. Keyword QUANT announces a quantity followed by its name, location of the values, the number of values and the file position of the quantity.

6.1.4.1 Example

```
> fsimapper scan FLOTHERMMAPLIB ./BGA_detailed.flofea
MESH Root_Assembly
PART   V    encapsulant         7168      5766
PART   V    die                 1200       722
PART   V    solder_ball_1_1       20         4
PART   V    solder_ball_1_2       20         4
QUANT TEMPERATURE ELEM          39475   2966255
```

In this example four volumes are addressed by 'V', they have the ids 1,2,3 and 4. In addition FloTHERM file does contain element based TEMPERATURE which can be read from file. One or more ids can be entered separated by whitespaces in the configuration file to obtain mapped quantity values.
In this example we now want to couple all volume parts of the model and therefore have to add the following lines to the configuration file:

```
sourceFileType
FLOTHERMMAPLIB
sourceMeshPartIds
1 2 3 4
```

6.1.5 ANSYS

The MpCCI FSIMapper scanner for ANSYS does not work with native solver files so that original `".cdb"` and `".db"` files need to be converted with `fsimapper convert ANSYS model.db` (see ▷ 5.2.4 ANSYS Scanner and Converter ◁ for details) producing intermediate .ml format related to FloTHERM and FloEFD format described in ▷ 6.1.4 FloTHERM ◁ and ▷ 6.1.3 FloEFD ◁.

(!) Please note ANSYS requirements and model preparation section of ▷ 5.2 ANSYS ◁.

The scanner for ANSYS export files can be selected by its name ANSYS in the generic MpCCI FSIMapper scan command, e.g.

```
> fsimapper scan ANSYS file_name.ml
```

The resulting output of the scanner has a simple tabular format with just the keywords MESH or PART. MESH assigns the name of the model to be read in. PART specifies 'S' or 'V', if it is a surface or a volume, the part name as well as two integers specifying the total number of nodes and elements of the part. Parts are implicitly numbered by their order of appearance starting at index 1.

6.1.5.1 Example

```
> fsimapper scan ANSYS ./plate.ml
MESH ANSYSDB2MAPLIB
PART   S   SHELLS__      5802      11600
PART   V   SOLIDS__      15565     76037
```

One shell or surface part (SHELLS__) addressed by 'S' and id 1 and one solid part (SOLIDS__) addressed by 'V' and with id 2 are defined in this ANSYS model.

A mapping of the surface component SHELLS__ would require to add

```
targetFileType
ANSYS
targetMeshPartIds
1
```

of the solid component SOLIDS__ would require to add

```
targetFileType
ANSYS
targetMeshPartIds
2
```
to the configuration file.

6.1.6 Abaqus

The part of the Abaqus model on which the quantity values are going to be mapped, has to be defined as a surface (with the keyword *SURFACE in the input deck) or in case of solid temperature mapping as a seperate element set (*ELSET) in the Abaqus model.

The Abaqus file scanner will list all defined elsets and surfaces found in the Abaqus input deck and assign unique positive ids for surfaces and negative ids for elsets.

For the MpCCI FSIMapper surfaces or elsets to be mapped need to be addressed in the configuration file.

Please see the following example in ▷ 6.1.6.1 Example ◁ how to determine a correct surface or elset id for a given input deck.

(!) If the scanned .inp file contains *INCLUDE directives for other parts of the Abaqus model stored in separate .inp files, the scanner also lists all surfaces found in the included input decks due to their appearance in the file flow.

6.1.6.1 Example

As an example the output of the Abaqus scanner for a simple input file "exampleAbaqus.inp" is shown below.

```
> fsimapper scan ABAQUS exampleAbaqus.inp
[FSIMapper] SET-1                elset    -1
[FSIMapper] SET-2                elset    -2
[FSIMapper] FLAP_SLAVE           surface   0
[FSIMapper] WALL_MASTER          surface   1
[FSIMapper] COUPLING_SURFACE     surface   2
```

Three surfaces (`FLAP_SLAVE`, `WALL_MASTER` and `COUPLING_SURFACE`) addressed by the global ids 0, 1 and 2 and two elsets (`SET-1` and `SET-2`) with ids -1 and -2 are defined in this Abaqus model.

One or more ids can be entered separated by whitespace in the configuration file to obtain mapped quantity values on the surfaces or elsets.

In this example we now want to couple the last surface for an Abaqus target model and therefore have to add the following lines to the configuration file:

```
targetFileType
ABAQUS
targetMeshPartIds
2
```

A mapping of temperature on elset `SET-1` would require to add

```
targetFileType
ABAQUS
targetMeshPartIds
-1
```

to the configuration file.

⊘ MpCCI FSIMapper does not support mixing elset and surface ids in targetMeshPartIds, e.g. a target model can either be defined by surface ids or elset ids.

6.1.7 EnSight Gold Case

The EnSight Gold Case scanner lists all scalar variables which are defined in the given .case file. At present the MpCCI FSIMapper reads all defined surfaces and volume parts together with user selected quantities. As in EnSight Gold Case format the quantity names may vary but denote the same physical meaning, the user has to select the quantities of interest by its name. The scanner process is split up into two steps. At first the given Case master file is scanned for available scalar quantities followed by geometric scan process which identifies and lists the part structures of containing geometries.

6.1.7.1 Example

As an example the output of the EnSight Gold Case scanner for a simple case file "exampleEnSight.case" is shown below.

```
> fsimapper scan ENSIGHT exampleEnSight.case
<CPLDATA>
# This file was automatically created for the MpCCI GUI.
```

```
#@ Code: "EnSight"
#@ Scan: "1"
#@ File: "exampleEnSight.case"
#@ Path: "..."
#@ Date: "..."
#@ User: "..."
#@ Host: "..."
#
# MpCCI relevant information:
#   Model dimensions  : unknown
#   Coordinate system : unknown
#   Solution type     : unknown
#   Load cases        : unknown
#   Unit system       : unknown
#   Precision         : unknown
#
    Total_Temperature         SCALAR      1
</CPLDATA>
<CPLDATA>
#@ Code: "EnSight"
#@ File: "exampleEnSight.case"
#@ Path: "..."
wall_1          S       2
inlet           S       3
wall_2          S       4
flexible_pin    S       5
outlet          S       6
Region_1        V       1
</CPLDATA>
```

The first "CPLDATA" block lists all defined scalar variables by name, the second block lists available volumetric 'V' and surface 'S' parts which can be used with MpCCI FSIMapper. To inform the MpCCI FSIMapper to handle variable "Total_Temperature" as physical temperature quantity, the user has to select it by name in the GUI or add

```
EXPORT_1002_TEMPERATURE
Total_Temperature
```

to the configuration file when using batch mode. Currently supported keywords in the MpCCI FSIMapper configuration file are:

```
EXPORT_1002_TEMPERATURE
Name_Of_Temperature
EXPORT_1003_WALLHTCOEFF
Name_Of_Heat_Transfer_Coefficient
EXPORT_1004_WALLHEATFLUX
Name_Of_Heat_Flux
EXPORT_1010_PRESSURE
Name_Of_Pressure
```

As the EnSight Gold Case reader can be used as source or target model we additionally have to add

```
sourceFileType
ENSIGHTGOLD
```

to the MpCCI FSIMapper configuration file if the case EnSight is used as source format

```
targetFileType
ENSIGHTGOLD
```
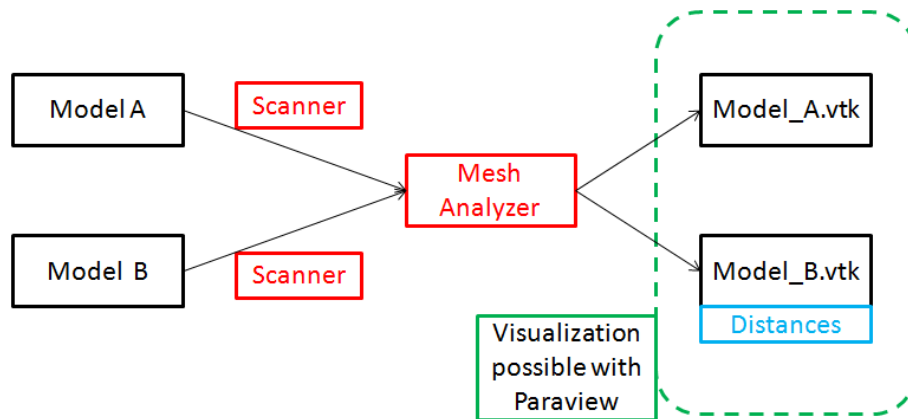
otherwise.

6.2 Comparing Geometries

Figure 17: The workflow of the MpCCI FSIMapper in batch usage when comparing two geometries

It is possible to use the MpCCI FSIMapper to compare two geometries defined in two model files `source` and `target`.

For the comparison of geometries the nodal distance as well as the association distance are computed. As can be seen in Figure 17 these distances are written to the .vtk file containing the target geometry.

For each node of the target mesh the nodal distance is the distance to the nearest point of the source mesh. The association distance is the distance between the point of the target mesh and the closest (associated) element of the source mesh. If the point lies "in" a source element, this would mean that the association distance is 0.

If the MpCCI FSIMapper is used to compare two geometries rather than map actual quantity values, this has to be set in the configuration file via the option sourceQuantityMode or targetQuantityMode (one of them is sufficient):

```
sourceQuantityMode
GEOMETRYCOMPARE
or
targetQuantityMode
GEOMETRYCOMPARE
```

Furthermore, the configuration file needs to contain all relevant information about the source and target file types, units and part ids. More information concerning these parameters is given in ▷ 6.5 Configuration File ◁, that gives an introduction to the MpCCI FSIMapper configuration file. The ids for the different mesh parts can be obtained by the respective mesh scanners.

With the appropriate configuration file the MpCCI FSIMapper can now be started by the command line:
```
> fsimapper batch configFile source target
```
(see ▷ 3 MpCCI FSIMapper Command ◁)

6.3 Mapping Quantities

The mapping of quantity values can be done with the same command line call as the comparison of geometries:

```
> fsimapper batch configFile source [source_quant] target
```

The configuration file contains all relevant parameters that are necessary for the mapping process. The different keywords and values are described in ▷ 6.5 Configuration File ◁.

The quantity file `source_quant` is an optional argument. It is necessary if the names of the model and quantity files are not the same, for example if the FLUENT.cas and .dat files do not have the same base name.

The MpCCI FSIMapper will print some output to the console while the mapping is in progress. After the mapping the result files can be found in the working directory.

6.4 Files Written by the MpCCI FSIMapper

As mentioned before MpCCI FSIMapper exports

- mapped quantities as input files for CSM codes and

- both source and target meshes as .vtk files.

The results with the mapped quantities can be used for an Abaqus (▷ 5.1 Abaqus ◁), ANSYS or MSC Nastran (▷ 5.13 MSC Nastran ◁) analysis.

Files in the open source format .vtk can be visualized by the open source viewer "Paraview" (www.paraview.org). Each of the two files (named for example "abaqus-mapped.vtk" and "fluent.vtk") contains the mesh definition as well as the original or mapped quantity data respectively. For the most common case of thermal mapping from e.g. FLUENT to Abaqus this means, that the FLUENT file contains the temperature, film temperature and wall heat flux values while the Abaqus file holds the mapped values of the film temperature and the wall heat coefficient.

6.5 Configuration File

The configuration file contains all parameters about file formats, geometry parts, quantities and mapping or orphan filling algorithms that define the complete process. The file is in a simple ASCII format. Different keywords define certain parameters. In the line following the keyword the respective value for the keyword is given. Lines starting with a "#" sign will be ignored.

```
#Example of configuration format structure
keyword_1
value
keyword_2
value
keyword_3
value_1 value_2 ... value_n
```

Information about the geometry or quantities have to be given for the source and the target models respectively.

Several different units are supported. Length, time, mass and temperature units can be specified for the source and the target model. The unit of pressure is expected to fit to the given mass, length and time units: $\text{Pa} = \frac{\text{kg}}{\text{m}\,\text{s}^2}$, for example, would be the matching pressure unit to kg, m and s. If the two models use different units all necessary conversions are done by the MpCCI FSIMapper.

Minimum required configuration file parameters:

```
#define source format type
sourceFileType

#define target format type
targetFileType

#define quantity mode for mapping
sourceQuantityMode

#define source parts
sourceMeshPartIds
id_1 ... id_n
#define target parts
targetMeshPartIds
id_1 ... id_m
#use default mapping algorithm and parameters
optionUseNeighborhoodDefaultParameters
true
```

All possible keywords and values are listed in the following table.

Keyword	Value
sourceFileType	FLUENT → Code FLUENT
	FLOTHERMMAPLIB → Code FloTHERM
	FLOEFDMAPLIB → Code FloEFD
	ENSIGHTGOLD → EnSight Case Gold
targetFileType	ABAQUS → Code Abaqus
	ANSYS → Code ANSYS
	NASTRAN → Code MSC Nastran
	ENSIGHTGOLD → EnSight Case Gold
sourceQuantityMode	ABSPRESSURE
	⇒ Read and map pressure
	STATICPRESSURE
	⇒ Read and map static pressure
	OVERPRESSURE
	⇒ Read and map overpressure
	PRESSURETRANSIENT
	⇒ Read and map transient pressure
	PRESSUREHARMONIC
	⇒ Read and map harmonic pressure
	FILMTEMPHTC
	⇒ Read and map surface film temperature and htc

	FILMTEMPUDM0
	⇒ Read and map surface film temperature and htc stored in UDM0 (only FLUENT)
	FILMTEMP
	⇒ Read and map surface film temperature
	TEMPERATURE
	⇒ Read and map volumetric temperature
	TEMPERATURETRANSIENT
	⇒ Read and map transient temperature
	WALLHEATFLUX
	⇒ Read and map surface heat flux
	FORCE
	⇒ Read and map surface forces
	FORCETRANSIENT
	⇒ Read and map transient surface forces
	FORCEHARMONIC
	⇒ Read and map harmonic surface forces
	FORCEDENSITY
	⇒ Read and map surface force density
	FORCEDENSITYTRANSIENT
	⇒ Read and map transient surface force density
	GEOMETRYCOMPARE
	⇒ Compare both geometries
	default: NONE
targetQuantityMode	GEOMETRYCOMPARE
	default: NONE
sourceMeshPartIds targetMeshPartIds	the zone ids of source and target mesh from the file scanner separated by whitespace
sourceMeshLengthUnit targetMeshLengthUnit	m, ft, in, cm or mm default: m
sourceTemperatureUnit targetTemperatureUnit	K or C default: K
sourceMeshMassUnit targetMeshMassUnit	kg, t or g default: kg
sourceMeshTimeUnit targetMeshTimeUnit	s or ms default: s
targetQuantityLocation	NODE, ELEMENT or DEFAULT default: DEFAULT = same location as source
optionMapAlgoType	type=shapefct type=weightedelement type=nearest type=conservative default: type=shapefct
optionUseNeighborhoodDefaultParameters	true or false default: false
optionNeighborhoodSearch	list of parameters [targetMeshLengthUnit unit] necessary for the neighborhood search: normaldistance, numberofnodes and radius
optionMultiplicity	value that increases neighborhood searching range default: 1

orphanFillType	type=distance
	type=average
	type=nearest
	type=none
	default: type=distance
fillOrphansValue	value that is assigned to orphaned nodes or elements.
	default: 0
optionComputeRotativeAverage	true or false
	default: false
rotationAngle	angle in angular degree
rotationAxisBeginPoint	x–, y– and z–coordinate of starting point separated by whitespace
rotationAxisEndPoint	x–, y– and z–coordinate of end point separated by whitespace
temperatureScaleValue	instead of several simulations with different temperatures it is also possible to scale temperature values by a certain factor e.g. having done a simulation with 500K and wanting to map a simulation with 600K the necessary scaling value would be 600/500 = 1.2
EXPORT_1002_TEMPERATURE	Assign the quantity name to be regarded as temperature
EXPORT_1003_WALLHTCOEFF	Assign the quantity name to be regarded as wall heat transfer coefficient
EXPORT_1004_WALLHEATFLUX	Assign the quantity name to be regarded as wall heat flux
EXPORT_1010_PRESSURE	Assign the quantity name to be regarded as pressure
EXPORT_1016_FORCE	Assign the quantity name to be regarded as force

A detailed description of the mapping algorithms and the concept of orphan filling can be found in ▷ 7.1 Numerical Methods ◁. The meaning of the parameters for the different mapping algorithms is also explained in this chapter in ▷ 7.1.2 Parameters for the Mapping ◁. The easiest solution for many applications might be to use the default parameters.

6.6 Example for a Configuration File

Here you find an example for an MpCCI FSIMapper configuration file. Lines starting with the "#" sign will be ignored. This example is a mapping from a FLUENT mesh (with unit system "m") to an Abaqus mesh (with unit system "mm").

The FLUENT film temperature (in K) is read from the .dat file. The temperature is written for an Abaqus simulation in ° C. The target quantity location is element–based and the shape function mapping algorithm is used with default parameters. The orphaned elements are going to be filled with 0.0. The optionComputeRotativeAverage keyword is set to false.

```
sourceFileType
```

```
FLUENT

targetFileType
ABAQUS

sourceQuantityMode
FILMTEMPHTC

targetQuantityMode
NONE

sourceMeshPartIds
136 146 114

targetMeshPartIds
1

sourceMeshLengthUnit
m

targetMeshLengthUnit
mm

sourceTemperatureUnit
K

targetTemperatureUnit
C

sourceMeshMassUnit
kg

targetMeshMassUnit
kg

sourceMeshTimeUnit
s

targetMeshTimeUnit
s

targetQuantityLocation  # location of quantities on Abaqus mesh
ELEMENT

# mapping algorithm type option
optionMapAlgoType        # mapping algorithm type option
type=shapefct

# neighborhood search options
optionUseNeighborhoodDefaultParameters
true
```

```
# fill orphans with a distance, average or nearest method
orphanFillType
type=distance

# fill-value for orphaned entities
fillOrphansValue
0.

optionComputeRotativeAverage
false

rotationAngle
30

rotationAxisBeginPoint
0 0 0

rotationAxisEndPoint
1 0 0

temperatureScaleValue
1.2
```

7 Theory Guide

7.1 Numerical Methods

7.1.1 Mapping Algorithms

The MpCCI FSIMapper currently offers three different algorithms to map quantity values between non-matching discretizations (see Figure 18). The distinctions and basic features of the mapping algorithms are:

Shape Function
  Shape function mapping simply interpolates a field using the shape functions. It can map linear functions exactly if linear elements are used; respectively quadratic functions need quadratic elements (i. e. elements with mid-nodes) to be mapped exactly.

Weighted Element
  Weighted element algorithm is designed to map element based values with quite small smoothing. It creates a relation between each target mesh node to a source mesh element and computes target element value by a distance weighted approach. Due to less smoothing it does recover higher gradients much better than the shape function approach.

Nearest
  The nearest mapping is based on a very simple neighborhood search algorithm. Every node in the target mesh receives the average value of the k closest nodes in the source mesh. The number k of nodes to be searched is given to the searching algorithm as a parameter. Alternatively, the average values from nodes lying within a certain radius can be used. Since for every node the k closest nodes can be found – no matter how far away they are positioned – this mapping algorithm never produces any orphaned nodes.

Conservative
  The conservative mapping is based on the nearest mapping algorithm and dedicated to integral quantities, e.g. forces. Every node in the target mesh receives a relative amount of the k closest nodes in the source mesh. Then, all unused nodes of source mesh distribute their information onto the k closest nodes of the target mesh.

  Since for every node the k closest nodes can be found – no matter how far away they are positioned – this mapping algorithm never produces any orphaned nodes.

Figure 18: Mapping algorithms and parameters in MpCCI FSIMapper

7.1.2 Parameters for the Mapping

The mapping algorithms need different parameters to control the mapping process (see Figure 18).

According to the chosen mapping algorithm the following parameters need to be set:

Default

No further parameter values are needed. The MpCCI FSIMapper uses default values for the chosen algorithm. These values depend on the source and target discretizations. For many applications, where the distances between the two geometries is quite small, this should yield good mapping results.

SurfaceMapping – Normal Distance

The maximal distance which is still accepted in normal direction. This is needed for "Shape Function" and "Weighted Element".
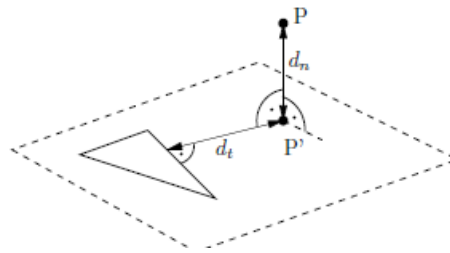

Figure 19: Normal distance d_n computation for element to node neighborhood used in shape function and weighted element mapping

A reasonable choice for the distance parameter might be

$$\text{Normal Distance} \quad \max(\text{SourceMaxEdgeLength}, \text{TargetMaxEdgeLength})$$

NNearest – Number of Nodes

This parameter is only needed for the neighborhood search used by the "Nearest" mapping algorithm. It gives the number of close nodes that are searched for every target point in the source mesh. A sensible choice for this parameter is e. g. 4.

Radial – Radius

This parameter is only needed for the neighborhood search used by the "Nearest" mapping algorithm. As an alternative to the search for a certain number of close nodes, the specification of this parameter leads to a search for all source nodes lying in a circle around the target point with the given radius.

Conservative – Maximal Distance

This parameter is only needed for the neighborhood search used by the "Conservative" mapping algorithm. If the target model is only a subset of the source model, a conservative transfer of non-matching areas to the subset will lead to unwanted cummulation of loads. Thus, the maximal search distance of data transfer can be limited.

Multiplicity

This parameter is needed for "Shape Function" and "Weighted Element" mapping algorithms. There it is used for the iterative neighborhood search leading to a more robust algorithm. The default value is 1.0. If a bigger value is chosen by the user, the searching radius (distance up to which a point is still regarded as a neighbor) will be enlarged (doubled) step-by-step until the given multiplicity of the starting radius is reached.

These suggestions for parameter settings should work fine with well matching geometries.

The parameter `Normal Distance` is not needed for the "Nearest" mapping algorithm. If the quantity values are mapped using this algorithm only one of the two parameters `NNearest – Number of Nodes` or `Radial – Radius` needs to be given.
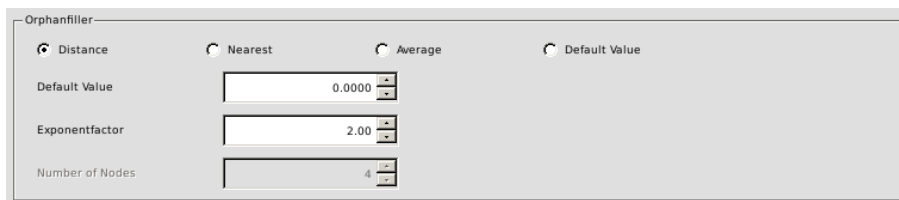
7.1.3 Orphan Filling

Figure 20: Methods and parameters for orphan filling in MpCCI FSIMapper

The MpCCI FSIMapper offers four different methods to fill the orphaned nodes or elements of a mesh. These are the nodes or elements of the target mesh, that did not receive any quantity values during the mapping process. This can happen if the two geometries do not match closely, e.g. if the target geometry contains model parts that are not present in the source geometry.

The easiest way to assign a value to the orphaned nodes is to define a constant "orphan fill value". Alternatively one of four available methods for orphan filling can be used. The distance filler can take an exponent value as an optional parameter.

During the filling process the orphan filler builds a "hole list". Most orphans will be located in one of these holes, but it is possible for orphans to not appear in the hole list. For example free floating orphaned regions of a mesh, that are not connected to any other part, do not form a hole because there are no border points from where values might be interpolated.

It is possible to hand one optional parameter to the distance type orphan filler. For all other fillers the specification of the type is all that is needed. The four different available orphan filling methods are:

Distance
> This orphan filler computes the distance to every border point for each orphaned node in a hole. The border values are then scaled with the reciprocal distance and added up for every orphan. One optional parameter can be handed to this filler: The `Exponentfactor`. This parameter determines to which power the distance - in the denominator - is taken. The default value for the `Exponentfactor` is 2.0, achieving a scaling with the square distance.

Nearest
> The `Nearest` orphan filler divides the target mesh by mapped and orphaned points or elements into two parts. It then uses a nearest search algorithm finding the by `Number of Nodes` specified number of closest mapped values for an orphan and computes the orphan value by a distances weighted averaging. The default value for `Number of Nodes` is 4.

Average
> This simple algorithm assigns the average quantity value of the border points to every orphan. The quantity therefore is constant on every hole.

Default Value
> All orphans are going to receive the specified orphan fill value.

7.2 Geometry and Quantity Transformations

7.2.1 Fourier Transformation and Windowing

The Fourier transformation of the discrete transient signal $(s(t_k))_{k=0}^{L-1}$ with L equidistant time steps of size τ results in frequency dependent complex data. The maximal frequency, which can be extracted from the signal is $F = 1/2\tau$. The size of the discrete frequency steps is determined by the reversed time period of s.

If the transient signal does not build a full period, Windowing is recommended. This feature multiplies the signal with a so-called window function w. The Fourier transformation is then applied to the signal $(s(t_k) \cdot w(k))_{k=0}^{L-1}$.

Several predefined window functions[1] are offered:

- **Hann**

$$w(k) = 0.5 - 0.5 \cdot \cos\left(\frac{2\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

- **Hamming**

$$w(k) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

- **Blackman**

$$w(k) = 0.42 - 0.5 \cdot \cos\left(\frac{2\pi k}{L-1}\right) + 0.08 \cdot \cos\left(\frac{4\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

- **Blackman-Harris**

$$w(k) = a_0 - a_1 \cos\left(\frac{2\pi k}{L-1}\right) + a_2 \cos\left(\frac{4\pi k}{L-1}\right) - a_3 \cos\left(\frac{6\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

  with $a_0 = 0.35875, \quad a_1 = 0.48829, \quad a_2 = 0.14128, \quad a_3 = 0.01168$.

- **Blackman-Nuttall**

$$w(k) = a_0 - a_1 \cos\left(\frac{2\pi k}{L-1}\right) + a_2 \cos\left(\frac{4\pi k}{L-1}\right) - a_3 \cos\left(\frac{6\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

  with $a_0 = 0.3635819, \quad a_1 = 0.4891775, \quad a_2 = 0.1365995, \quad a_3 = 0.0106411$.

- **Barlett**

$$w(k) = \frac{2}{L-1} \cdot \left(\frac{L-1}{2} - \left|k - \frac{L-1}{2}\right|\right), \quad k = 0, \ldots, L-1$$

- **Barlett-Hann**

$$w(k) = 0.62 - 0.48 \cdot \left|\frac{k}{L-1} - 0.5\right| - 0.38 \cdot \cos\left(\frac{2\pi k}{L-1}\right), \quad k = 0, \ldots, L-1$$

- **Cosinus**

$$w(k) = \cos\left(\frac{\pi k}{L-1} - \frac{\pi}{2}\right), \quad k = 0, \ldots, L-1$$

---

[1]taken from `https://de.wikipedia.org/wiki/Fensterfunktion`, July 5, 2016

- **Tukey**

$$w(k) = \begin{cases} \frac{1}{2}\left[1 + \cos(\frac{2\pi k}{\alpha(L-1)} - \pi)\right], & \text{if} \quad 0 \le k \le \frac{\alpha(L-1)}{2} \\ 1, & \text{if} \quad \frac{\alpha(L-1)}{2} \le k \le (L-1)(1-\frac{\alpha}{2}) \\ \frac{1}{2}\left[1 + \cos(\frac{2\pi k}{\alpha(L-1)} - \pi(\frac{2}{\alpha} - 1))\right], & \text{if} \quad (L-1)(1-\frac{\alpha}{2}) \le k \le (L-1) \end{cases}$$

with $\alpha = 0.5$.

- **Lanczos**

$$w(k) = \text{sinc}\left(\frac{2k}{L-1} - 1\right), \quad k = 0, \dots, L-1$$

The user can define his own window function as a one-columned file, where the L window function values $w(k)$, $k = 0, \dots, L-1$ are defined.

7.2.2 Cyclic Symmetry of Quantities

The mapping of quantities present on a cyclic symmetric source model to a full target model requires to build previously the corresponding full source mesh with data.

The (complex) data on cyclic symmetric models can exhibit a special periodicity, defined by the so-called (forward/backward) nodal diameter $ND \in \mathbb{N}, 0 \le ND \le \lfloor n/2 \rfloor$ (or cyclic symmetry mode).

It defines the temporal phase lag between the data of two neighboring sections as $\frac{2\pi \cdot ND}{n}$ (forward) or $\frac{-2\pi \cdot ND}{n}$ (backward). A nodal diameter 0 means that the data are the same on each section ("constant"). For an even number of cyclic symmetric sections, the nodal diameter $n/2$ indicates, that the data changes sign from section to section ("alternating"). An illustration of the meaning of nodal diameters is given in Figure 21 for the example of a periodic model with $n = 8$.
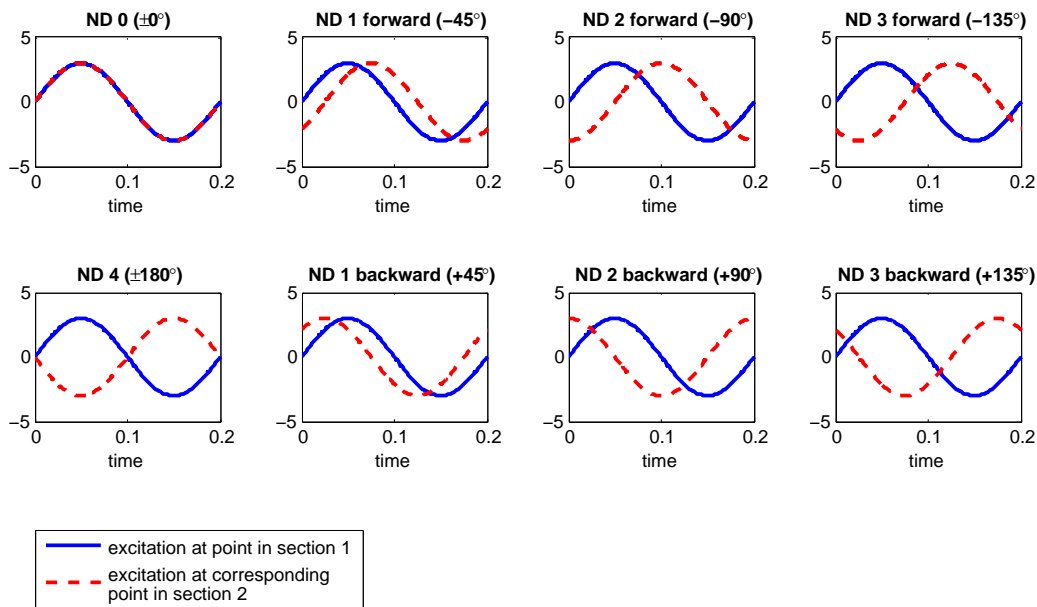
Figure 21: Phase lag between two neighboring sections for different nodal diameters on a 8-periodic model

Quantities of cyclic symmetric structures can exhibit the following nodal diameters in MpCCI FSIMapper:

Analysis Type	n even/odd	Available Nodal Diameters
Static		0
Transient	n even	$0, n/2$
	n odd	0
Harmonic	n even	$0, 1, \ldots, n/2$
	n odd	$0, 1, \ldots, (n-1)/2$

The nodal diameters $0 < ND < n/2$, which only occur for harmonic cyclic symmetric data, are referred to as "paired". Those quantities are described with both real and imaginary part and both are used simultaneously in order to create the data of the full model.

The data's cyclic symmetry mode, which is needed to calculate the quantity's values on the full model, is determined by the periodicities in the considered system.

For instance, periodic (rotating) disturbances produced by the k-th harmonic of an m-periodic part lead to a forward excitation nodal diameter $ND \in \{0, 1, \ldots, \lfloor n/2 \rfloor\}$, if there exists a positive integer a with $ND = -(k \cdot m - a \cdot n)$. If there is a positive integer b with $ND = k \cdot m - b \cdot n$ the excitation shape is in backward mode. The cyclic symmetry axis needs to be chosen, such that the relative rotation speed is positive around this axis. Please refer to **?** for further information.

ⓘ Only with a properly selected (forward/backward) nodal diameter, the data will be spatially continuous over the periodic boundaries in the corresponding full model. This is the prerequisite of a correct mapping to a full model or a model with different section shape, as shown in Figure 8.

ⓘ The direction of the axis of cyclic symmetry and the forward/backward excitation mode are closely connected through the following: if the excitation shape is defined by a forward resp. backward nodal diameter ND with given cyclic symmetry axis $\overrightarrow{a}$, then it is equivalent to define it as backward resp. forward nodal diameter ND with axis $-\overrightarrow{a}$.

ⓘ For harmonic analyses, please check how the structural solver defines the nodal diameter/cyclic symmetry mode of the excitation. If the solver only knows the forward excitation mode (e.g. Abaqus) use the previous note in order to switch between backward and forward by changing the sign of the cyclic axis in the structural solver.

8 Tutorial

The tutorials demonstrate the use and the capabilities of FSIMapper for some special applications.

The general process of mapping data from a result file ("source") to a structural model ("target") is visualized in Figure 1. First, the simulation of the source model is performed. The source result file (native or special export depending on simulation software) and a prepared mesh of the target model are read by MpCCI FSIMapper. With application dependent settings, which need to be defined in the GUI or for batch usage in the config file, the mapping is done. As a result MpCCI FSIMapper creates a file(s) which contains the mapped quantities for the target mesh in the target code format. The file(s) is included in the target simulation in order to define the loading applied by the mapped source data.

8.1 Mapping of Electromagnetic Forces for Transient and Harmonic Analyses

8.1.1 Problem Description

In this tutorial, a two pole generator, called TEAM-24, is studied with respect to its vibration behavior. A transient electromagnetic simulation is performed using Infolytica MagNet where the resulting forces acting on the stator are calculated. By a transient mapping and a subsequent Fourier transformation of the data, the loading for a frequency response analysis is created. Here, MSC Nastran is used as target simulation code.

Rotor and stator exhibit two poles each (cf. Figure 22), where the rotor turns 1° per ms, i. e. with a frequency of $2.\overline{7}$ Hz. Due to the symmetry of the problem the model is truncated at its symmetry plane.
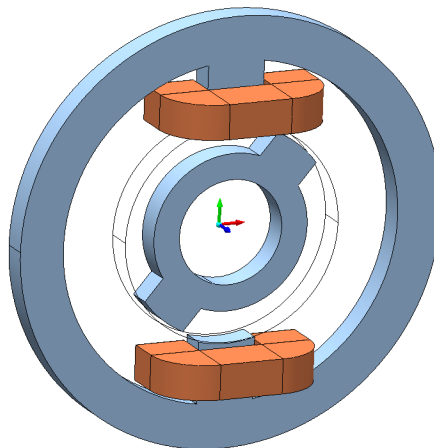
Figure 22: TEAM-24 model in MagNet

The files concerning this tutorial are located in

- MpCCI FSIMapper as part of the MpCCI installation:
  "<*MpCCI_home*>/tutorial/FSIMapper/Team24-Stator_ElectromagneticForces"

- MpCCI FSIMapper standalone installation:
  "<*MpCCIFSIMapper_home*>/tutorial/Team24-Stator_ElectromagneticForces"

Before mapping, copy the files to your working directory.

8.1.2 Source Result File

In MagNet the TEAM-24 generator is simulated by the Solve→Transient 3D with Motion... solver. The transient solver options (Solve→Set Transient Options...) are set such that a constant time step is used (cf. Figure 23) which is obligatory for the Fourier transformation in MpCCI FSIMapper. It is sufficient to simulate only one half turn (180° in 180 ms) because of the cyclic symmetry with two poles.
The magnitude of the force density at 23 ms calculated by MagNet is shown in Figure 24.
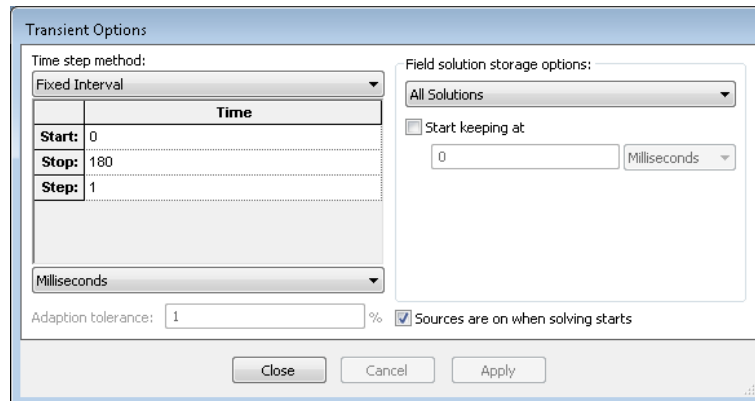
Figure 23: Options for transient MagNet simulation
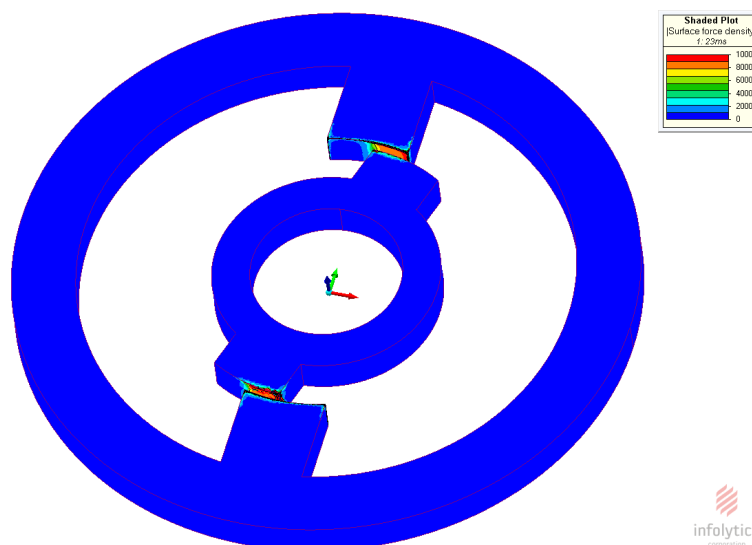
Figure 24: Surface force density at 23 ms computed by MagNet

For the use in MpCCI FSIMapper the surface force density has to be exported in MagNet using a special export called "Exporter for MpCCI". The procedure is the following (cf. Figure 25):

- Select in the object tree the component whose surface load is to be exported (here "Stator")
- Open Extensions→Exporter for MpCCI
- Enter as "Period" 180

- Push the $\boxed{\text{Start}}$ button
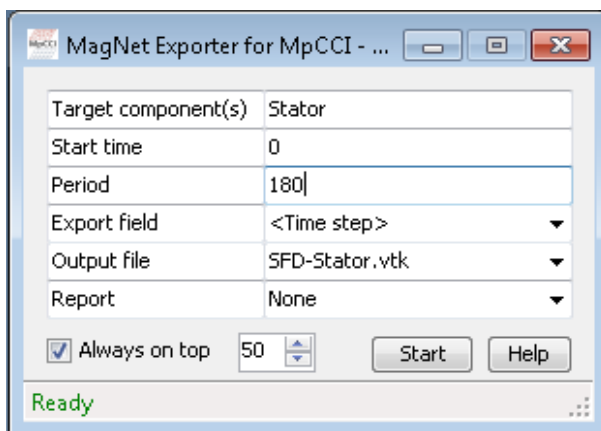
Figure 25: MagNet's "Exporter for MpCCI"

The exporter creates the file `"SFD-Stator.vtk"` including all time steps between 0 ms and 179 ms.

ⓘ For a Fourier transformation without windowing (as it is implemented in MpCCI FSIMapper) it is necessary to use transient data points which can be stacked one behind the other in order to create continously the data for the following periods, as shown in the first picture in Figure 26.

As the data at 180 ms is exactly the same as at 0 ms and the time steps 0 to 179 prescribe one period (in the sense of the first picture in Figure 26), the prerequisites for a correct Fourier transformation are fulfilled.

ⓘ In the .vtk file the SI unit system is used.
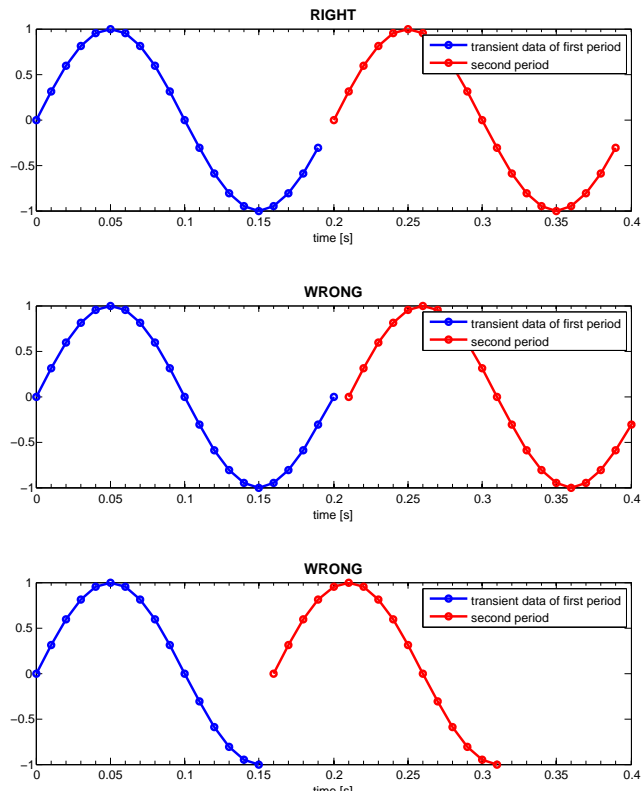
Figure 26: How to define transient data used by the Fourier transformation (without windowing). The blue dots are the data points provided for the mapping (here, a discrete sine-signal of frequency 5 Hz and sampling time step 0.01 s). The red dots are generated by stacking the data behind the end of the provided time signal. Only the first definition is correct

8.1.3 Target Mesh File

The MSC Nastran target mesh models the stator by first order hexa and penta elements, see Figure 27. The mapping surface is defined by shell elements (green) which use the same nodes as the volume elements (blue).

(a) Overview

(b) Zoom to Interior
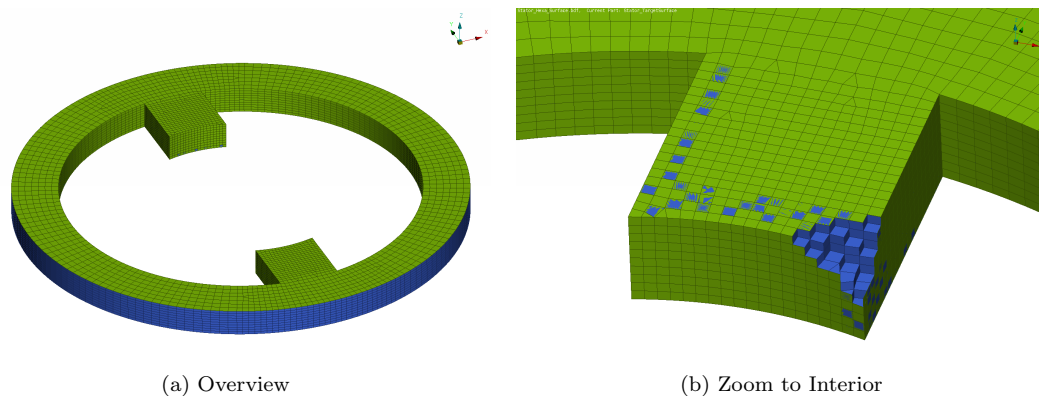
Figure 27: Target structural mesh (blue) with shell elements (green) defining the mapping surface

The unit system used is *mm-t-s* which leads to force unit *N*.

8.1.4 Mapping

Open the MpCCI FSIMapper GUI and change the settings in the "How to map/Transient" panel as follows (shown in Figure 28):

1. Select MagNet in the source file type drop-down menu and choose the "SFD-Stator.vtk" by pressing the ⬚... button

2. Select "SFD-Stator.vtk" from the parts list

3. Set the source unit system to SI

4. Select the force density "SFD" in the "Force" variable drop-down menu

5. For the target simulation code select MSC Nastran and the mesh file "Stator_Hexa_Surface.bdf"

6. Select the surface abbreviated by "S" with MSC Nastran PSHELL PID 2 in the parts box.

7. Set the unit system of the target model to *mm-t-s*.

This configuration maps for each time step the electromagnetical surface force density to the stator. In this tutorial we want the mapped transient force density to get transformed using Fourier analysis. Therefore, check "Apply Fourier Transformation" in the "Result" panel in order to output a corresponding include file that defines the loading for linear vibration analyses, cf. Figure 29.

Pressing the Map button starts the mapping process.
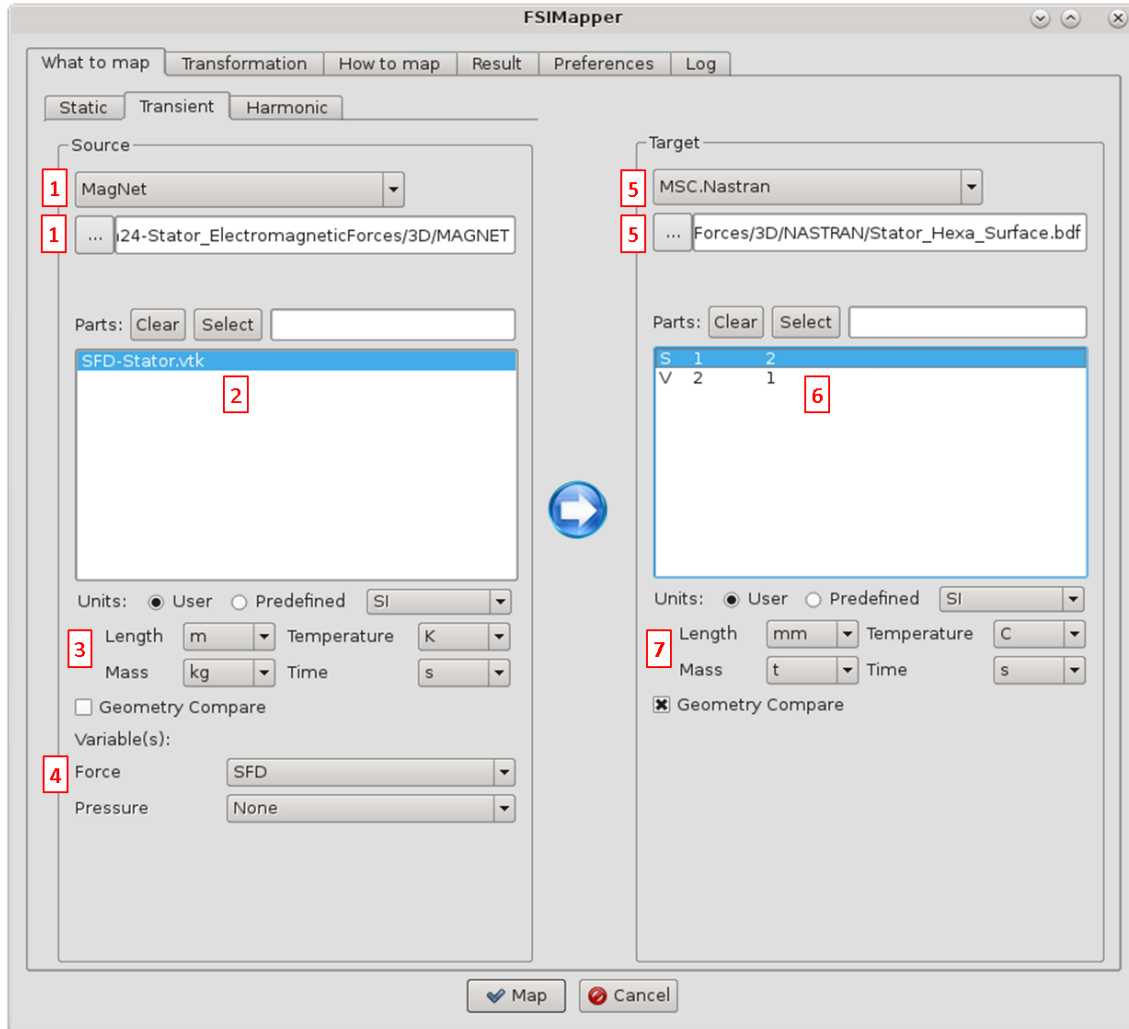
Figure 28: The "What to map" panel
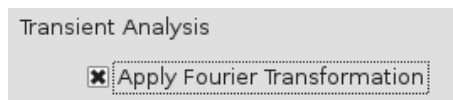
Figure 29: The "Apply Fourier Transformation" option in the "Result" panel

When the mapping finishes, the MpCCI Visualizer shows the mapping results for each time step (cf. Figure 30 at 23 ms). Due to the coarser target mesh, the mapped surface force density differs at first glance from the source data. But integrating both densities will lead to the same total force.
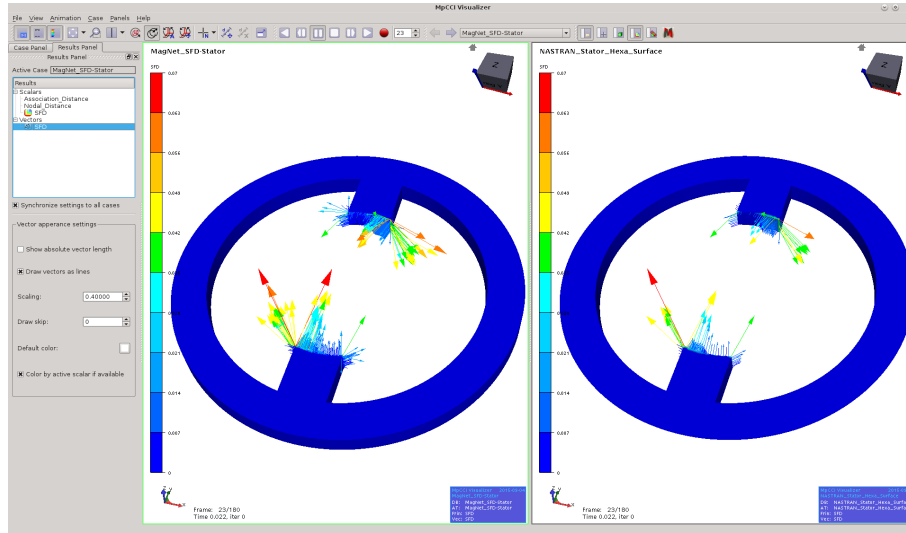

Figure 30: Transient mapping result shown in MpCCI Visualizer at 23 ms

With a sampling time step of $\tau = 0.001\,\mathrm{s}$ and a number of samples $L = 180$ the Fourier transformation results in the frequency range from

$$0\,\mathrm{Hz} \quad \mathrm{to} \quad \frac{\lfloor L/2 \rfloor}{\tau \cdot L} = 500\,\mathrm{Hz} \quad \mathrm{in} \quad \frac{1}{\tau \cdot L} = 5.\overline{5}\,\mathrm{Hz} \quad \mathrm{steps}$$

MpCCI FSIMapper creates a MSC Nastran include file "Stator_Hexa_Surface-mapped_FreqRespForce.bdf" where the Fourier transformation results are applied as complex forces to the nodes building the mapping surface. It also defines the frequencies for the frequency response analysis:

```
$
$ DLOAD = 600
$ FREQ  = 601
$
$-------2-------3-------4-------5-------6-------7-------8-------9-------0-------
FREQ*                601 0.000000000E+00 5.555555556E+00 1.111111111E+01
*        1.666666667E+01 2.222222222E+01 ...
...
$
$-------2-------3-------4-------5-------6-------7-------8-------9-------0-------
$ Node 1, DOF 1
RLOAD1        600        1                        1        2
DAREA           1        1        11.00E+00
$ Real part for each frequency
TABLED1*               1
*
*        0.000000000E+00 5.050313394E-03 5.555555556E+00-1.127201827E-03
*        1.111111111E+01-2.292065593E-03 1.666666667E+01 1.449636200E-03
```

```
*         2.222222222E+01 2.616820479E-04 ...
...
$ Imaginary part for each frequency
TABLED1*               2
*
*         0.000000000E+00 0.000000000E+00 5.555555556E+00-3.482561041E-03
*         1.111111111E+01 1.660795053E-03 1.666666667E+01 1.071458938E-03
*         2.222222222E+01-8.023966910E-04 ...
...
```

Moreover, MpCCI FSIMapper creates the file `"Stator_Hexa_Surface-mapped_FreqRespForce.ccvx"` in order to post-process the Fourier transformed forces. It is opened in MpCCI Visualizer and shows the corresponding transient force fluctuations (at 10 pseudo time steps) for each frequency. Open Panels→Results Panel and select in the "Results" list e. g. "Frequency_005.555555555" as "Scalar" and "Vector" quantity, see Figure 31a and 31b. Switch off View→Perspective Projection and select in the Panels→Settings Panel the outline representation as shown in Figure 31c.

Pressing the ▷ button animates the 10 pseudo time steps. With the value given in File→Preferences→Maximum animation FPS the animation speed is set. In this way one can visualize the frequency content of the transient signal, see Figure 32. Frequency $5.\bar{5}$ Hz has the highest influence in the transient force progression which is twice the rotation speed (resulting from the two poles). The higher orders play a decreasing role and the excitation shapes get more and more complicated.
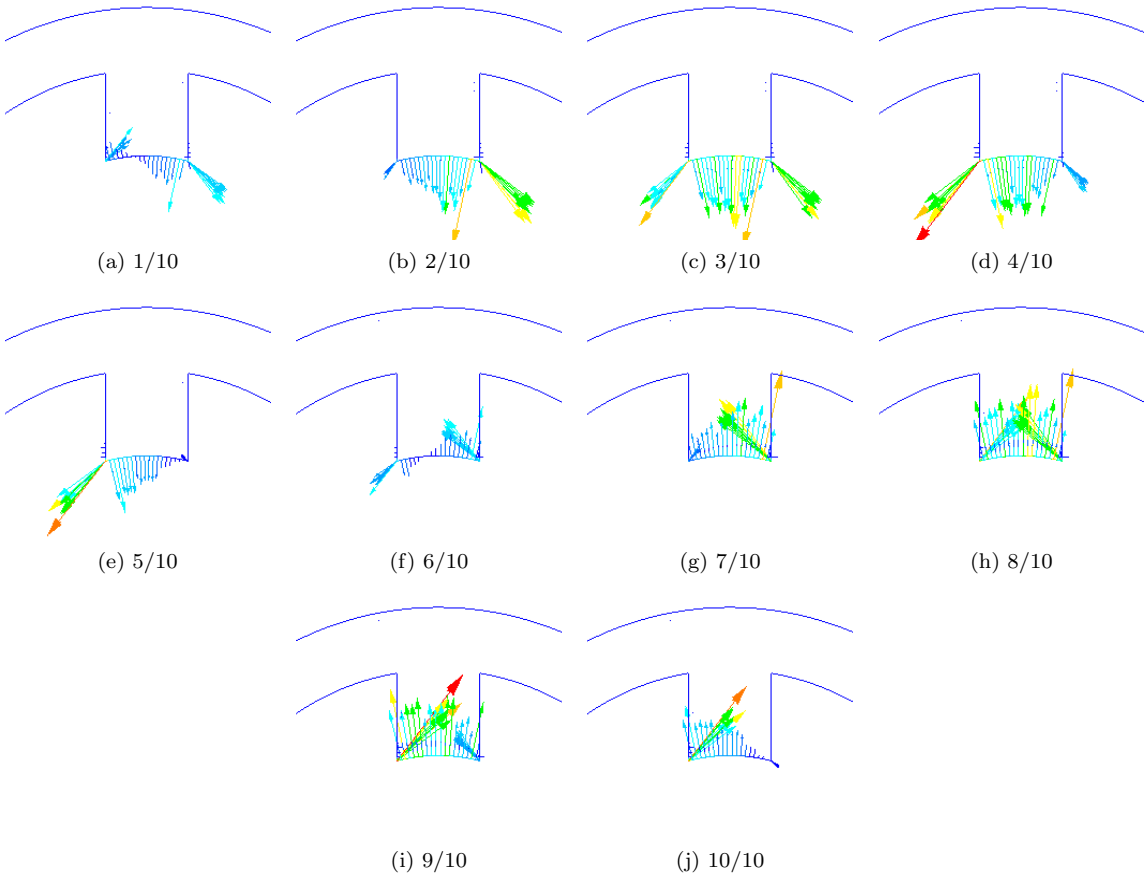
(a) "Results Panel", "Scalar" quantity   (b) "Results Panel", "Vector" quantity   (c) "Settings Panel"
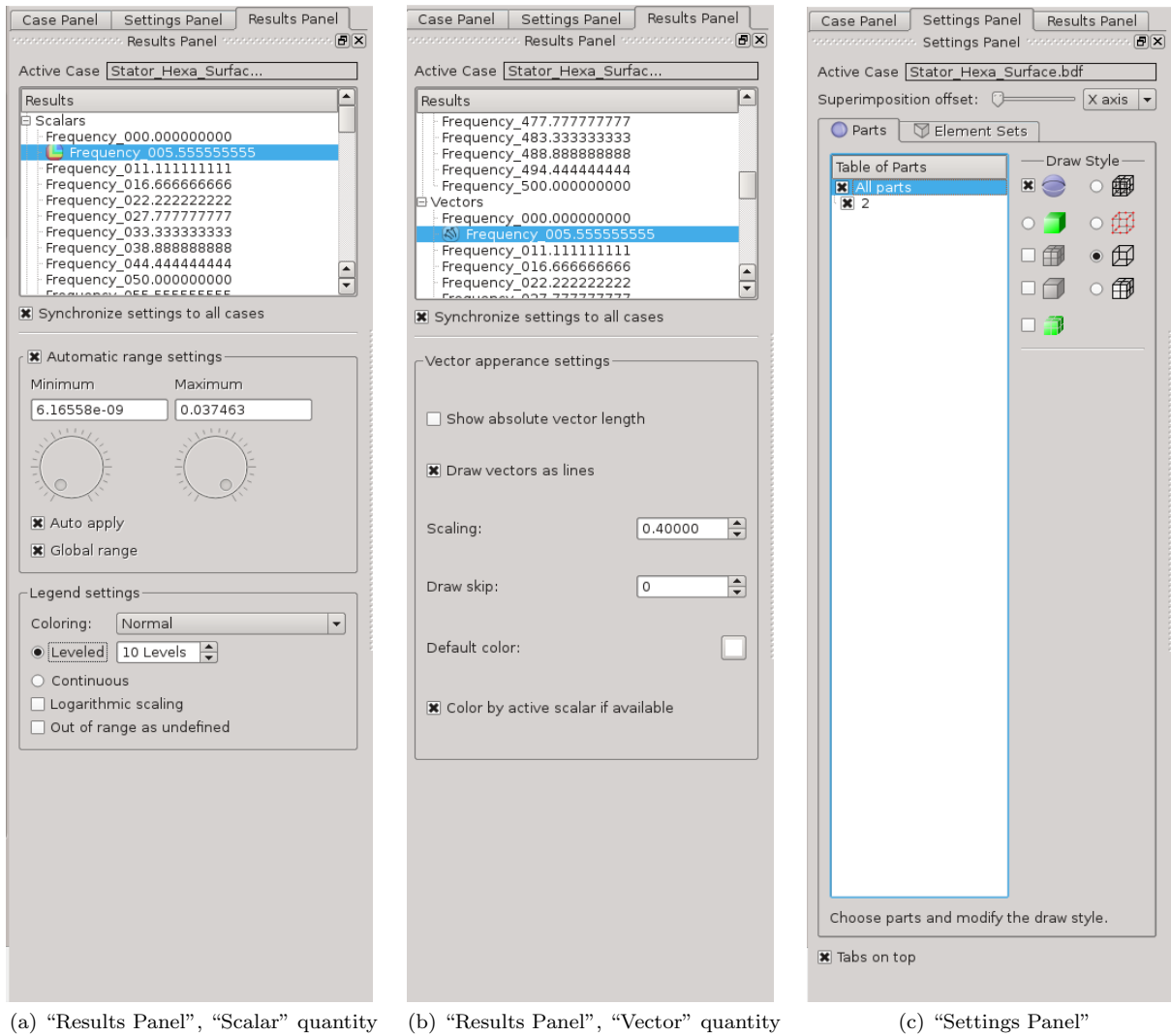
Figure 31: MpCCI Visualizer settings

(a) 1/10   (b) 2/10   (c) 3/10   (d) 4/10

(e) 5/10   (f) 6/10   (g) 7/10   (h) 8/10

(i) 9/10   (j) 10/10

Figure 32: Corresponding transient force excitation at $5.\overline{5}\,\mathrm{Hz}$ generated for 10 pseudo time steps

8.1.5 Target Simulation

MpCCI FSIMapper exported a MSC Nastran include file containing the nodal force excitation resulting from a Fourier transformation of the electromagnetic surface force density calculated by MagNet. It is located in the same folder as the MSC Nastran input deck. The load case definition in MSC Nastran uses the file in the following way:

```
$
SOL 111 $ Modal Frequency Response
CEND
$
TITLE = Team-24_Stator3D_FreqResp
$
$
  METHOD = 150
  SPC = 1
  FREQ = 601
  DLOAD = 600
  DISP(PLOT) = ALL
  STRESS(PLOT) = ALL
  RESVEC = NO
  ECHO = NONE
$
$
BEGIN BULK
$
$-------2-------3-------4-------5-------6-------7-------8-------9-------0------
EIGRL   150    -0.1    1250
$
INCLUDE 'Stator_Hexa.bdf'
INCLUDE 'Stator_Hexa_Surface-mapped_FreqRespForce.bdf'
$
$-------2-------3-------4-------5-------6-------7-------8-------9-------0-------
$
PARAM, POST, -1
$
ENDDATA
```

For the simulation the MSC Nastran mesh "Stator_Hexa.bdf" is used, i. e. the volume mesh without the shells defining the mapping surface (keeping the same node ID's). Please refer to the MSC Nastran Dynamic Analysis User's Guide for more information about the frequency response analyses types SOL 108 and SOL 111.

(!) If the "Apply Fourier Transformation" option is not checked in the "Result" panel, then MpCCI FSIMapper creates the file "Stator_Hexa_Surface-mapped_TransientForce.bdf" which contains the mapped transient nodal forces. It can be used in SOL 109 and SOL 112 analyses.

8.2 Mapping of Harmonic Pressure Excitations in Turbomachinery

Periodic pressure fluctuations excite blade vibrations which can endanger the integrity of the whole system.

The harmonic CFD methods offer the possibility to simulate those excitations in an efficient way without the initial attack time. They approximate the transient steady state flow behavior as a superposition of the time-averaged state variable and periodic fluctuations around this mean value.

Due to the periodic nature of steady state turbomachinery flows the pressure oscillations can be splitted into frequency components (harmonics) which are multiples of the blade passing frequency.
The solution approach of the harmonic CFD methods is to determine the complex amplitudes (magnitude and phase lag) for each of the superposed harmonics. Thus, the methods do not step successively in time but work in frequency domain.

Structural frequency response analyses can use directly the complex pressure data of the harmonic CFD simulation. They define the flow-induced excitations at every considered frequency.

In this tutorial the mapping process is shown for the Harmonic Balance Method of STAR-CCM+ and the Nonlinear Harmonic Method of Fidelity Fine Turbo. Here, the concept of cyclic symmetry and nodal diameters is exemplarily discussed.

8.2.1 Using the Harmonic Balance Method of STAR-CCM+

8.2.1.1 Problem Description

In this tutorial the flow-induced blade vibration of the NASA Rotor37 in operation is simulated. It is based on the STAR-CCM+ tutorial "Harmonic Balance: Single Stage Periodic Flow" where the transient pressure fluctuation is approximated by three harmonics and the time-averaged pressure. Here, the target simulation code is Abaqus.

The mapping process presented here creates the loading for a structural frequency response analysis in the following steps:

1. map 1st harmonic pressure excitation

2. map 2nd harmonic pressure excitation

3. map 3rd harmonic pressure excitation

4. map time-averaged pressure

The tutorial also shows the mapping capability of MpCCI FSIMapper for periodic source and full target models.

Rotor37 has a geometrical periodicity of $n = 36$ blades. The upstream stator exhibits a periodicity of $m = 48$ blades; only the influence of this component is considered for this tutorial. The rotor rotates around the z-axis by $\omega = 17200\,\text{rpm} = 286.\overline{6}\,\text{Hz}$.

The files concerning this tutorial are located in

- MpCCI FSIMapper as part of the MpCCI installation:
  "$<MpCCI\_home>$/tutorial/FSIMapper/Rotor37"

- MpCCI FSIMapper standalone installation:
  "$<MpCCIFSIMapper\_home>$/tutorial/Rotor37"

Before mapping, copy the files to your working directory.

8.2.1.2 Source Result File

In the STAR-CCM+ tutorial "Harmonic Balance: Single Stage Periodic Flow" the flow conditions in the Rotor37 are simulated using the Harmonic Balance method. The transient solution is approximated by 3 harmonics (in STAR-CCM+ called "modes"), where the 48-bladed stator is modelled as Gaussian wake (so not geometrically). Only one blade passage is modelled of the rotor.

The magnitude of the first ($k = 1$) pressure harmonic at $k \cdot m \cdot \omega = 13760\,\mathrm{Hz}$ is shown in Figure 33. $m \cdot \omega$ refers to as blade passing frequency.
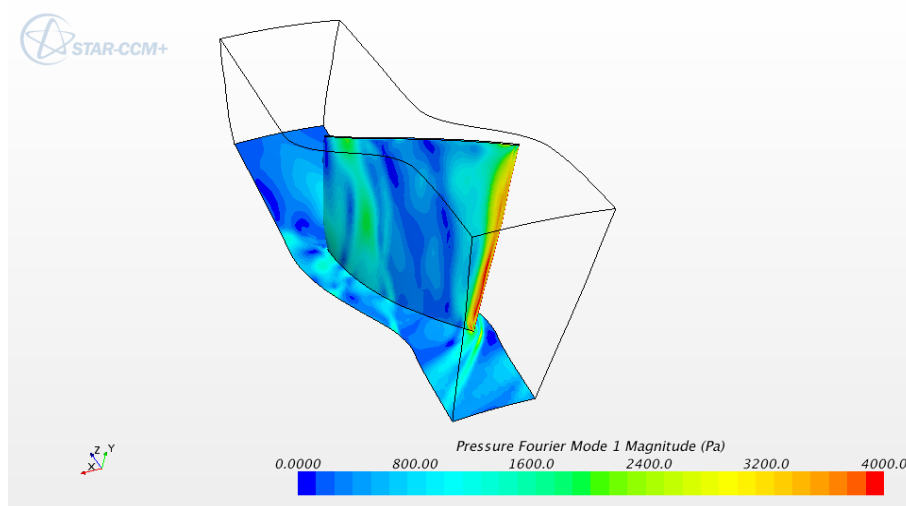
Figure 33: Magnitude of the first pressure harmonic

The harmonic results of the simulation are exported for the wetted surfaces into the EnSight Gold case format via File→Export… in STAR-CCM+, as shown in Figure 34.

The main resulting file is `"singleRowHarmonic.case"` which refers to the geometry file `"singleRowHarmonic00000.geo"`, the exported time-averaged (`"*.PressureFourierMode0"`) and harmonic pressures (`"*.PressureFourierMode*Real"`, `"*.PressureFourierMode*Imag"`).
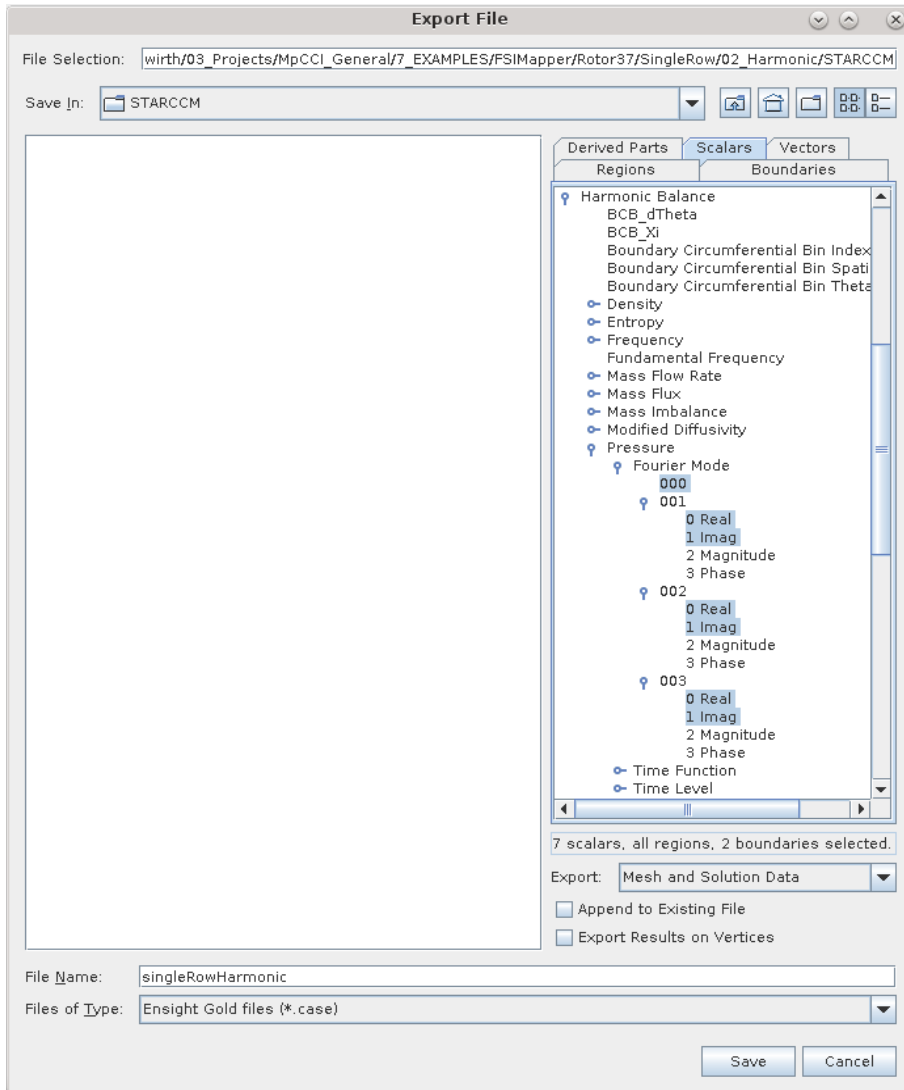
Figure 34: Export the harmonic results as EnSight Gold case file

8.2.1.3 Target Mesh File

The structural target mesh file `"rotor37_fine.inp"` comprises the whole rotor, as shown in Figure 35. Second order tetra elements were used.

(a) Overview   (b) Zoom
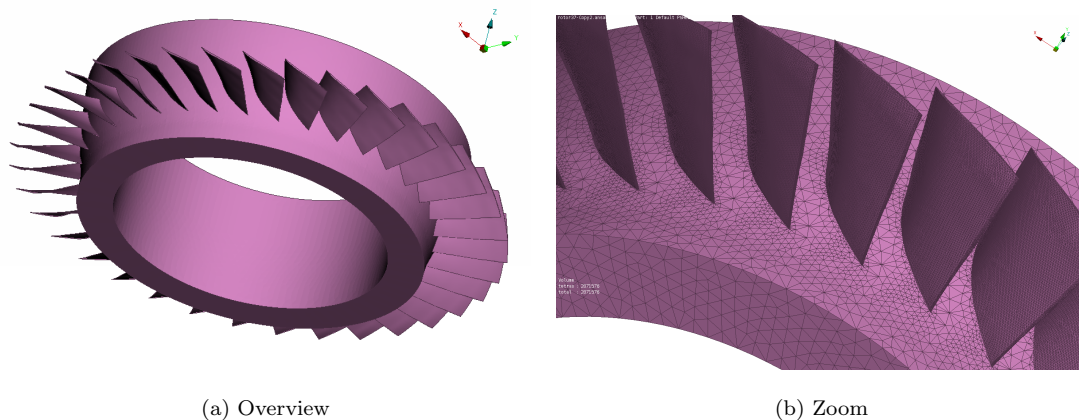
Figure 35: Target structural mesh of the full Rotor37

The mapping surface, i.e. the surface where the pressure excitation acts, needs to be defined. Internally, Abaqus uses the keyword `*SURFACE, NAME=WettedSurface, TYPE=ELEMENT`.

8.2.1.4 Mapping

First we map the first harmonic ($k = 1$) to the structural Abaqus model. Open the MpCCI FSIMapper GUI and change the settings in the "How to map/Harmonic" panel as follows (shown in Figure 36):

1. Select "EnSight Case" in the source file type drop-down menu and choose the file `"singleRowHarmonic.case"` by pressing the `...` button.

2. Select the wetted surfaces in the source model: "Hub_Surface_Blade_Row_1" and "Blade_Surface_Blade_Row_1"

3. Set the source unit system to SI

4. Select for the real part of the mapped quantity "PressureFourierMode1Real" and for the imaginary part "PressureFourierMode1Imag"

5. For the target simulation code select "Abaqus" and the mesh file `"rotor37_fine.inp"`

6. Select the surface "WettedSurface" in the parts box.

7. Set the unit system of the target model to *mm-t-s*.

Since the source mesh models only a periodic section of the rotor and the target mesh comprises the full rotor, a transformation is needed in order to generate matching geometries. For this purpose give the following information in the "Transformation/Geometry" subpanel (cf. Figure 37):

• Select the "User defined" transformation and check the option "Cyclic Symmetry"

• For the number of periodicities put in 36

• Define the cyclic symmetry axis via the two points $[0, 0, 0]$ and $[0, 0, 1]$, which is the positive z-axis
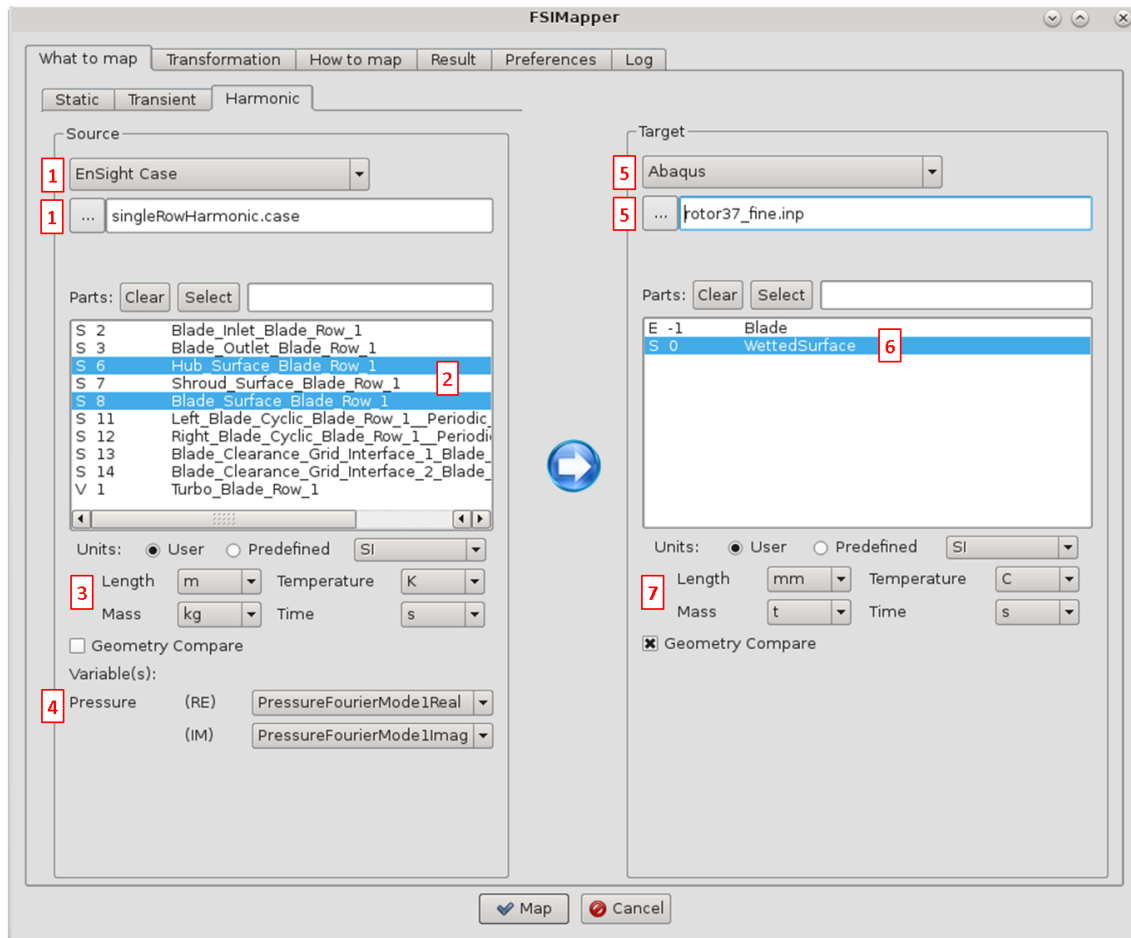
Figure 36: The "What to map" panel

Due to the 48-bladed stator, the excitation of the first harmonic is of a nodal diameter 12 periodicity in backward mode, i. e. the excitation travels in the opposite direction as the rotation sense.

This can be seen by the formula given in a note in ▷ 4.3 The "Transformation" Panel ◁:

$$ND = k \cdot m - b \cdot n = 1 \cdot 48 - b \cdot 36 = 12 \quad \text{with} \quad b = 1$$

Since the rotation is positive around the z-axis, which was given as cyclic symmetry axis in the "Geometry" subpanel, select the paired backward nodal diameter 12 in the "Quantity" subpanel.

Pressing the Map button starts the mapping process.
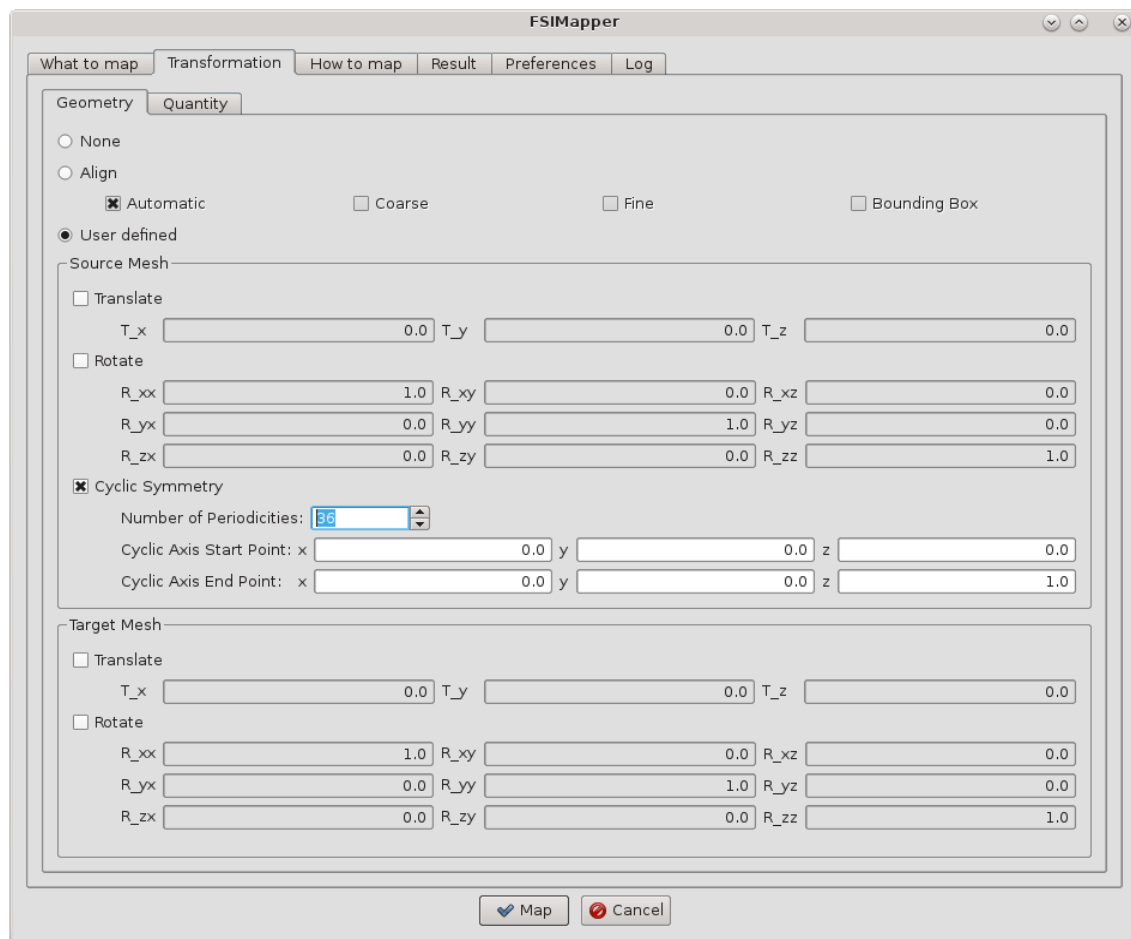
Figure 37: The "Transformation/Geometry" subpanel

When the mapping finishes, the MpCCI Visualizer shows the mapping results, cf. Figure 38.
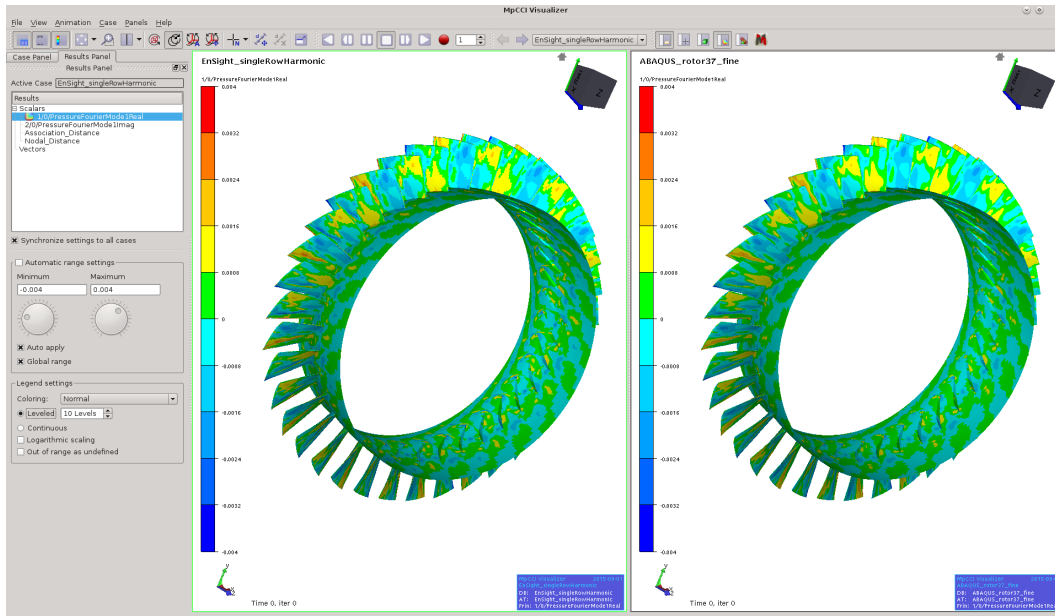
(a) Real Part
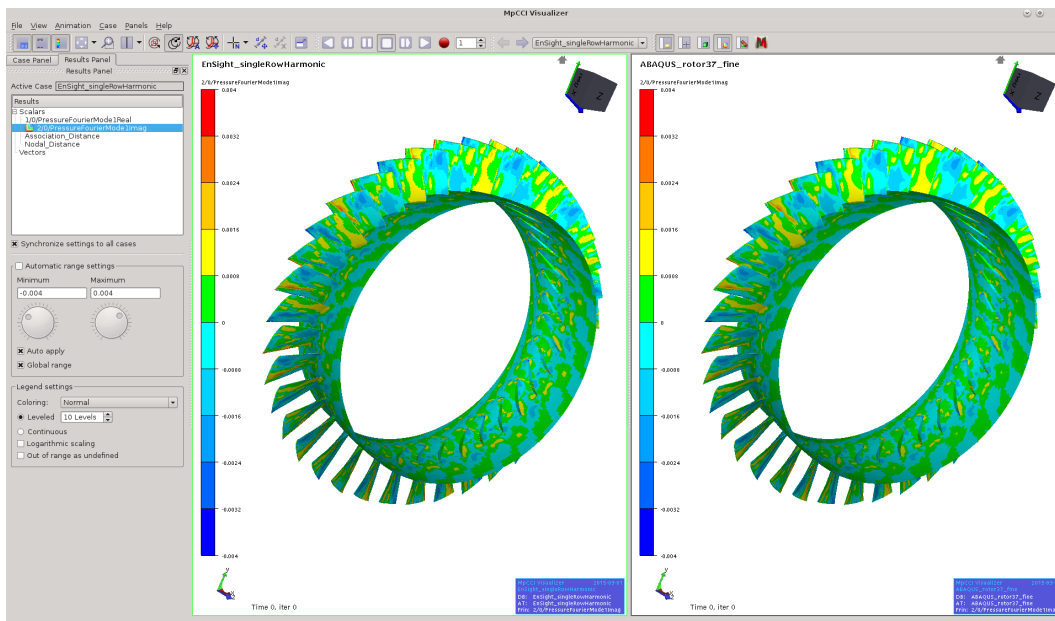
(b) Imaginary Part

Figure 38: Harmonic mapping result shown in MpCCI Visualizer

The periodic mesh and data were transformed by the given information in order to create the full source model and the corresponding data. The first harmonic complex pressure excitation has been revolved such that the data are spacially continuous over the periodic boundaries, see Figure 39. This is only the case for the correctly selected nodal diameter and excitation mode.
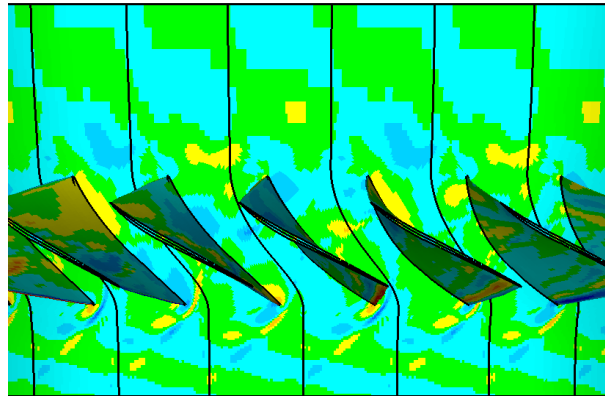
Figure 39: Cyclic Symmetry transformation creates continuous data over the periodic boundaries

MpCCI FSIMapper creates two Abaqus include files `"rotor37_fine-mapped_HarmonicPressure_RE.inc"` and
`"rotor37_fine-mapped_HarmonicPressure_IM.inc"` where the real and imaginary part of the pressure is applied to the elements building the wetted surfaces:

```
**
** Exported real part of surface pressure
** *DLOAD, REAL
205,  P1,  -0.00033113
473,  P1,  -0.00028541
10650,  P1,  0.00043067
10651,  P1,  0.00057767
10654,  P1,  -0.00037426
10656,  P1,  -0.00048984
...
```

```
**
** Exported imaginary part of surface pressure
** *DLOAD, IMAGINARY
205,  P1,  -0.00009977
473,  P1,  -0.00007182
10650,  P1,  -0.00031675
10651,  P1,  -0.00037232
10654,  P1,  0.00042306
10656,  P1,  0.00062463
...
```

Moreover, MpCCI FSIMapper creates the file `"rotor37_fine-mapped_HarmonicPressure.ccvx"`. It can be opened in MpCCI Visualizer and shows the corresponding transient pressure fluctuations for 18 pseudo time steps. Pressing the ▷ button animates them with a speed set in File→Preferences→Maximum animation FPS. In this way one can observe the backward travelling wave respective to the rotation sense.

ⓘ Before mapping the two remaining harmonics, rename the exported files as "<filename>_H1.<extension>".

For the second harmonic ($k = 2$) select the variables "PressureFourierMode2Real" and "PressureFourierMode2Imag". The nodal diameter 12 in forward mode has to be selected in the "Quantity" subpanel. Also rename the newly created files as "<*filename*>_H2.<*extension*>". Repeat the procedure for the third harmonic with a nodal diameter 0.

Also, we map the time-averaged pressure which represents the pressure level around which the harmonics oscillate. Switch in the "What to map" panel to "Static" and select the same file and parts as in the "Harmonic" subpanel, see Figure 40. Select the quantity "PressureFourierMode0" and map by pressing the Map button. For static analyses only the nodal diameter 0 is available which is selected here by default.
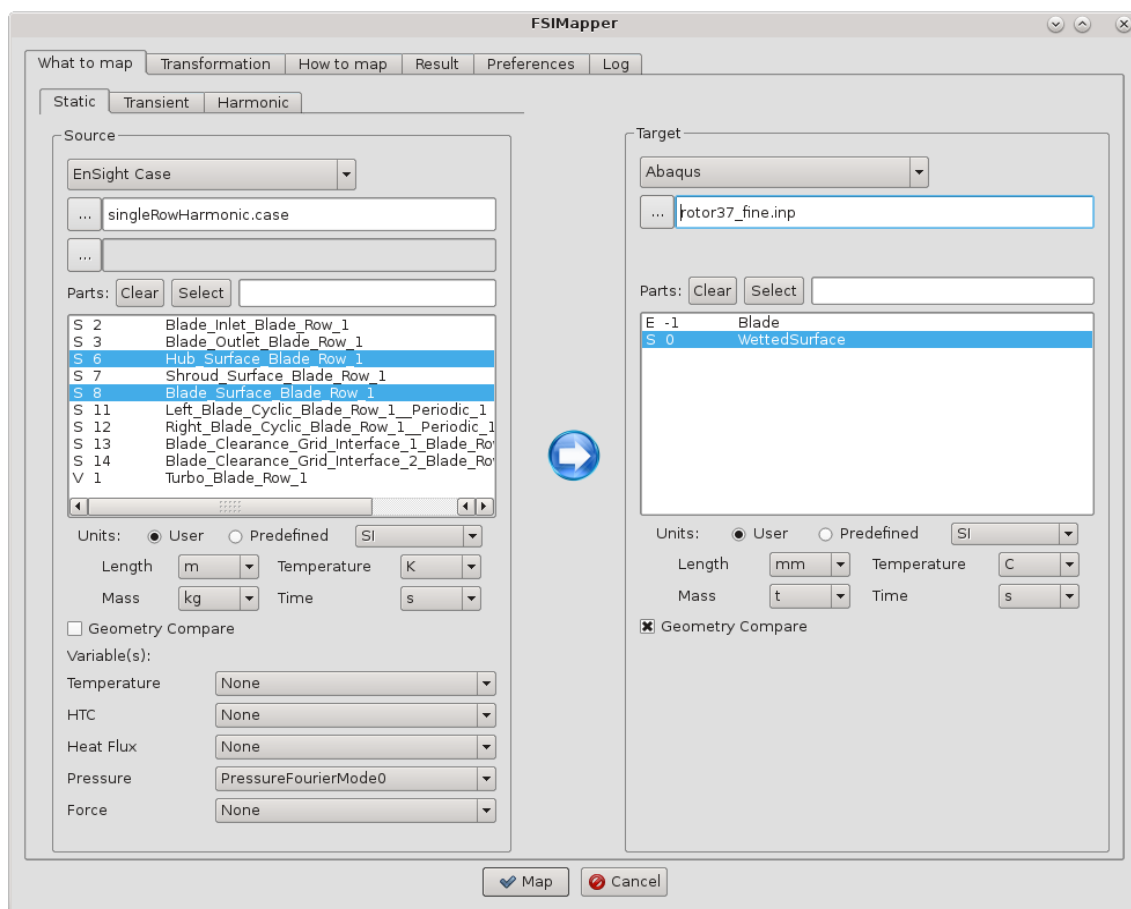

Figure 40: The "What to map/Static" panel

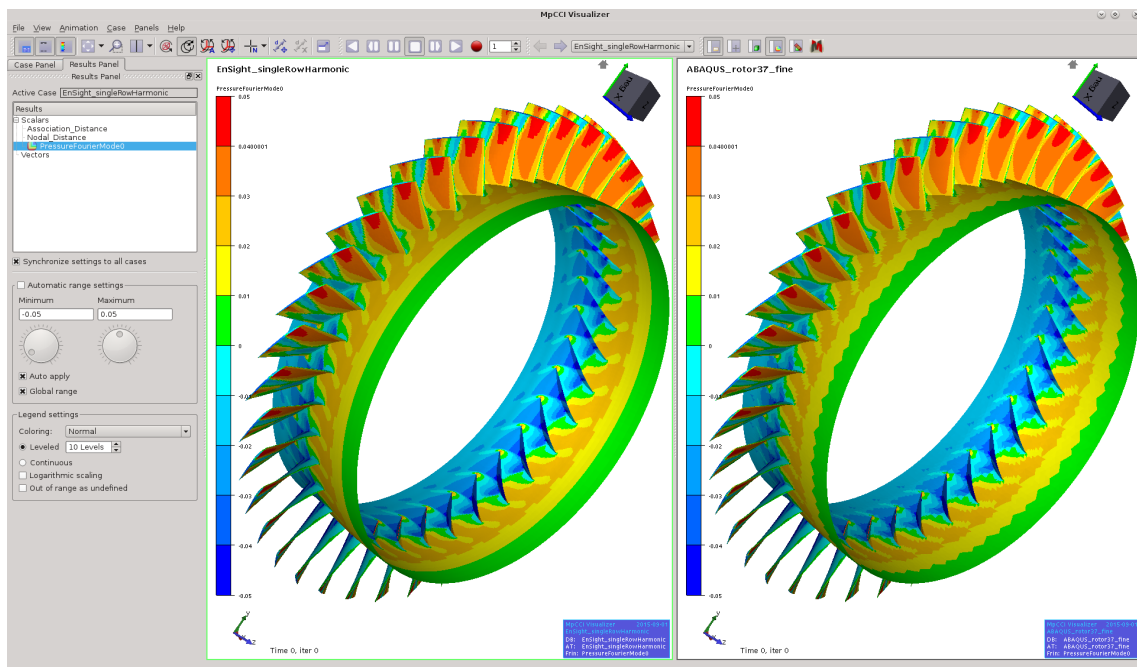The mapped static pressure is shown in Figure 41.

Figure 41: Mapping result of the time-averaged pressure

8.2.1.5 Target Simulation

MpCCI FSIMapper exported in summary for each of the tree harmonics two include files containing the real and the imaginary part of the pressure excitation. Moreover, an include file was exported which defines the mean/time-averaged loading. They are located in the same folder as the Abaqus input deck.

The load case definition in Abaqus uses these files in the following way:

```
**
*STEP, NAME=MeanPressure, NLGEOM=YES, INC=100000
**
*STATIC
0.1,1,1e-4,1
**
*DLOAD, OP=MOD
*INCLUDE, INPUT=rotor37_fine-mapped_Pressure.inc
**
*END STEP
**
**
**
*STEP, NAME=ModalAnalysis
**
*FREQUENCY, EIGENSOLVER=LANCZOS
 , 0, 100000, , , ,
**
*END STEP
**
**
**
*STEP, NAME=SSD_1H
**
*STEADY STATE DYNAMICS
13760, 0, , ,
**
*DLOAD, REAL
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_RE_H1.inc
**
*DLOAD, IMAGINARY
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_IM_H1.inc
**
*END STEP
**
**
**
*STEP, NAME=SSD_2H
**
*STEADY STATE DYNAMICS
27520, 0, , ,
**
*DLOAD, REAL
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_RE_H2.inc
**
*DLOAD, IMAGINARY
```

```
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_IM_H2.inc
**
*END STEP
**
**
**
*STEP, NAME=SSD_3H
**
*STEADY STATE DYNAMICS
41280, 0, , ,
**
*DLOAD, REAL
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_RE_H3.inc
**
*DLOAD, IMAGINARY
*INCLUDE, INPUT=rotor37_fine-mapped_HarmonicPressure_IM_H3.inc
**
*END STEP
**
```

Please check the Abaqus User's Keywords Reference Guide for the options of `*STEADY STATE DYNAMICS` concerning direct and modal solution methods.

(!) For the sake of completeness use in the first step also the centrifugal and the Coriolis forces (by the keyword `*DLOAD`).

(!) The damping in the system has fundamental influence to the magnitude of blade vibration, so it is recommended to define it.

Here, the full Rotor37 is simulated. Another possibility would have been to model it in Abaqus by a periodic section using the keyword `*CYCLIC SYMMETRY MODEL`. The data periodicity (nodal diameter) is given in Abaqus by the option `CYCLIC MODE=<nodal diameter>` in the keywords `*DLOAD` or `*CLOAD`.

(!) Since Abaqus only knows forward excitation modes, for backward excitations the cyclic symmetry axis (which was determined such that the rotation is positive) has to be turned around (only) for the definition in the Abaqus keyword `*CYCLIC SYMMETRY MODEL`.

8.2.2 Using the Nonlinear Harmonic Method of Fidelity Fine Turbo

8.2.2.1 Problem Description

In this tutorial the blade vibration of the axial turbine "Aachen" in operation is simulated. It is based on the Fidelity Fine Turbo tutorial "2. Axial Turbine" and the transient pressure fluctuation is approximated by three harmonics and the time-averaged pressure. Here, the target simulation code is Abaqus.

In this tutorial, the 1st harmonic pressure excitation is mapped to a structural frequency response analysis. The remaining harmonics can be mapped in the same way using the information given in ▷ 8.2.2.4 Mapping ◁. The time-averaged pressure is mapped in the "What to map/Static" panel.

The tutorial also shows the mapping capability of MpCCI FSIMapper for periodic source and target models with different section shapes.

The considered rotor in the "Aachen" turbine has a geometrical periodicity of $n = 41$ blades. The upstream and downstream stators exhibit a periodicity of $m = 36$ blades both. The rotor rotates around the z-axis by $\omega = -3500\,\mathrm{rpm} = -58.\overline{3}\,\mathrm{Hz}$.

The files concerning this tutorial are located in

- MpCCI FSIMapper as part of the MpCCI installation:
  "<*MpCCI_home*>/tutorial/FSIMapper/AxialTurbineAachen"
- MpCCI FSIMapper standalone installation:
  "<*MpCCIFSIMapper_home*>/tutorial/AxialTurbineAachen"

Before mapping, copy the files to your working directory.

8.2.2.2 Source Result File

The result of the static Fidelity Fine Turbo tutorial "2. Axial Turbine" is used as initial solution for a Nonlinear Harmonic simulation, where the transient solution is approximated by 3 harmonics in a rank-1 approach (equivalent to "Basic"). Only one blade passage is modelled for each of the three stages. For demonstration purposes the computation has been performed on grid level 1.

The magnitude of the first ($k = 1$) pressure harmonic at $k \cdot m \cdot \omega = 2100\,\mathrm{Hz}$ is shown in Figure 42. $m \cdot \omega$ refers to as blade passing frequency. Since the stators have the same number of blades, the influence of both is included in this harmonic.

The results of the simulation are comprised in the Fidelity Fine Turbo .cgns result file
"Tutorial2_NLH_Rank1_Har3_111.cgns".

(!) The harmonic data are only available in the volume mesh.

8.2.2.3 Target Mesh File

The structural target mesh file "aachenBlade.inp" comprises one section of the rotor, as shown in Figure 43. The section shape is different to the section in Fidelity Fine Turbo. First order hexa and penta elements were used.

The mapping surface, i. e. the surface where the pressure excitation acts, needs to be defined. Internally, Abaqus uses the keyword `*SURFACE, NAME=MappingSurface, TYPE=ELEMENT`.
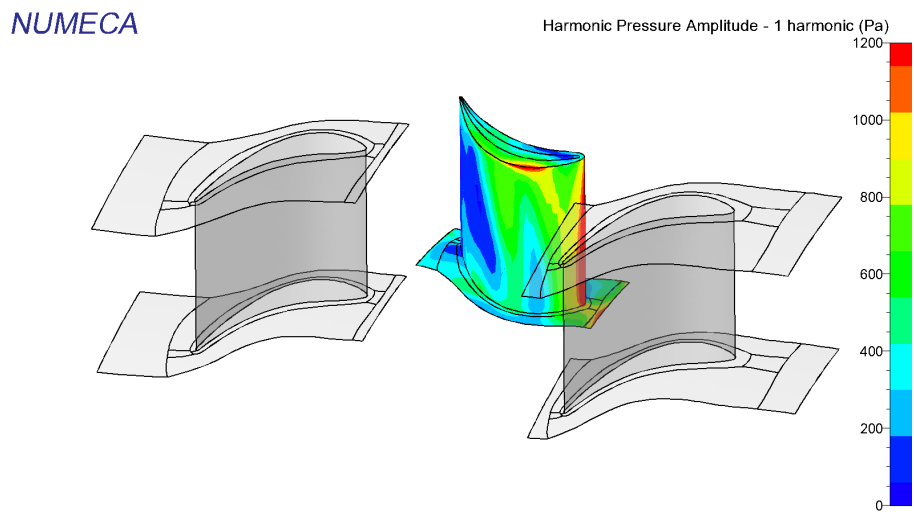
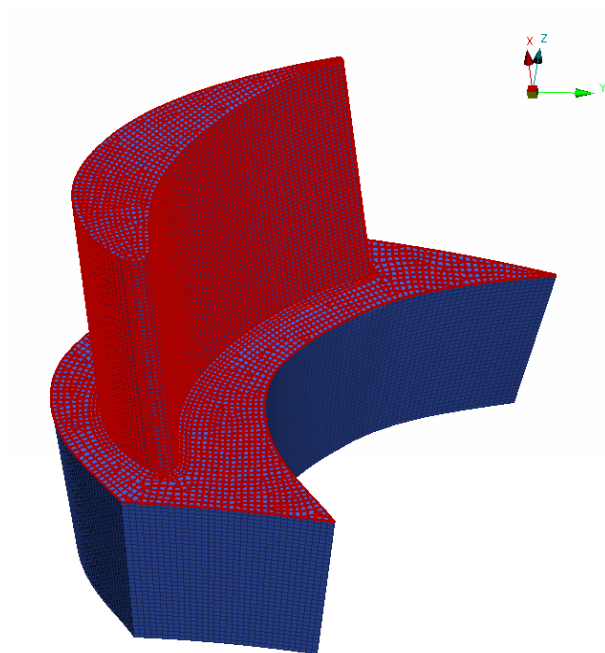Figure 42: Magnitude of the first pressure harmonic at 2100 Hz

Figure 43: Target structural mesh (blue) with wetted surface definition (red) of one periodic section of the axial turbine "Aachen"

8.2.2.4 Mapping

Here, we map only the first harmonic ($k = 1$) to the structural Abaqus model. The remaining two harmonics are handled equivalently ($k = 2$: forward nodal diameter 10, $k = 3$: forward nodal diameter 15).
Open the MpCCI FSIMapper GUI and change the settings in the "How to map/Harmonic" panel as follows (shown in Figure 44):

1. Select "FINE/Turbo" in the source file type drop-down menu and choose the file "Tutorial2_NLH_Rank1_Har3_111.cgns" by pressing the ... button.

2. Select the volume blocks (abbreviated by "V") which suround the wetted surface of the rotor in the source model: "domain8" to "domain16".

3. Set the source unit system to SI

4. Select for the real part of the mapped first harmonic pressure "ReP_1" and for the imaginary part "ImP_1"

5. For the target simulation code select "Abaqus" and the mesh file "aachenBlade.inp"

6. Select the surface "MappingSurface" in the parts box.

7. Set the target unit system to *mm-t-s*.

Since the source mesh models a periodic section, which does not match the target mesh section, a periodic transformation is needed in order to provide the data on the surfaces which are not covered by the source mesh. For this purpose give the following information in the "Transformation/Geometry" subpanel (cf. Figure 45):

- Select the "User defined" transformation and check the option "Cyclic Symmetry"

- For the number of periodicities put in 41

- Define the cyclic symmetry axis via the two points $[0, 0, 0]$ and $[0, 0, -1]$, which is the negative z-axis (which implies a positive rotation direction)

Due to the 36-bladed stators, the excitation of the first harmonic is of a nodal diameter 5 periodicity in forward mode, i.e. the excitation travels in the same direction as the rotation sense.
This can be seen by the formula given in a note in ▷ 4.3 The "Transformation" Panel ◁:

$$ND = -(k \cdot m - a \cdot n) = -(1 \cdot 36 - a \cdot 41) = 5 \quad \text{with} \quad a = 1$$

Since the rotation is positive around the negative z-axis, which was given as cyclic symmetry axis in the "Geometry" subpanel, select the paired forward nodal diameter 5 in the "Quantity" subpanel, see Figure 46.
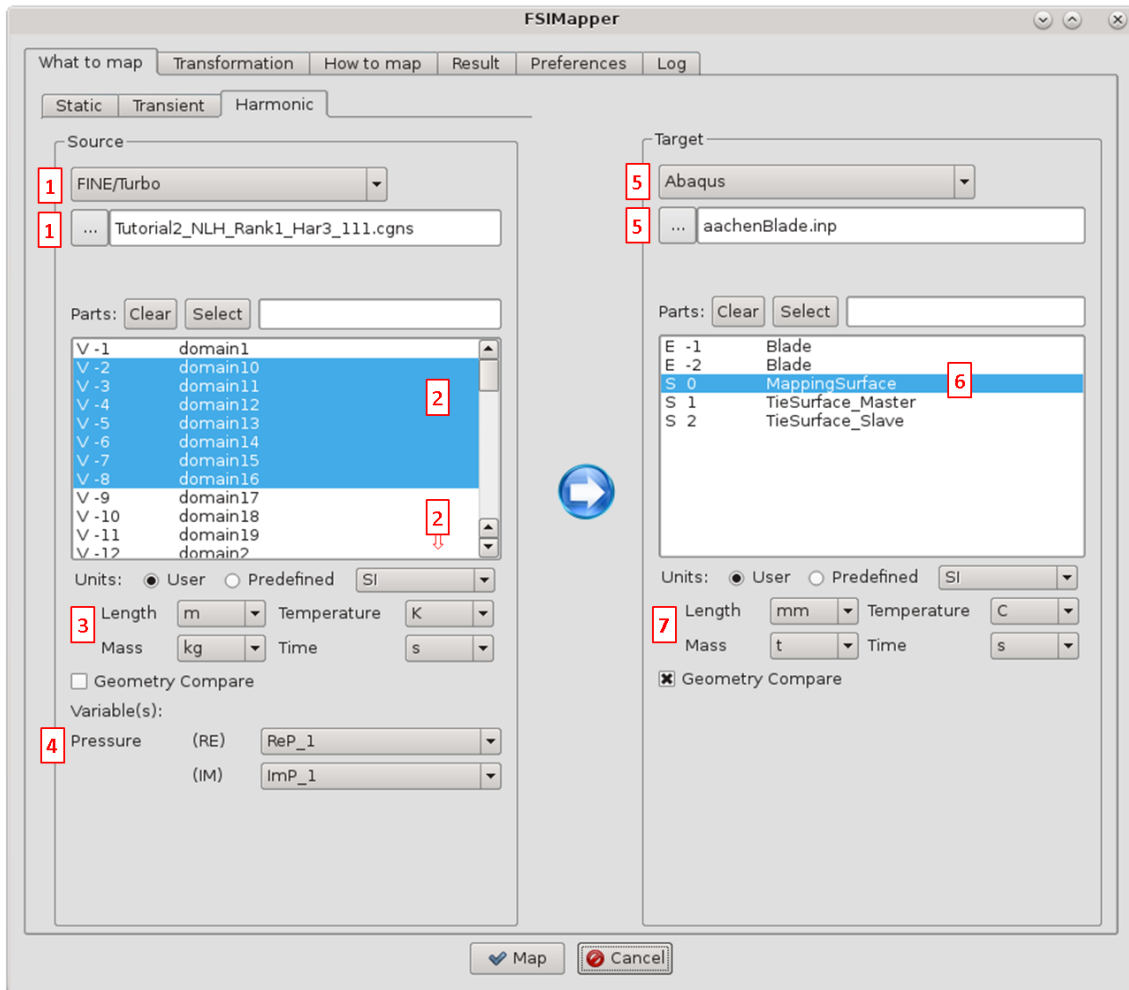
Pressing the Map button starts the mapping process.

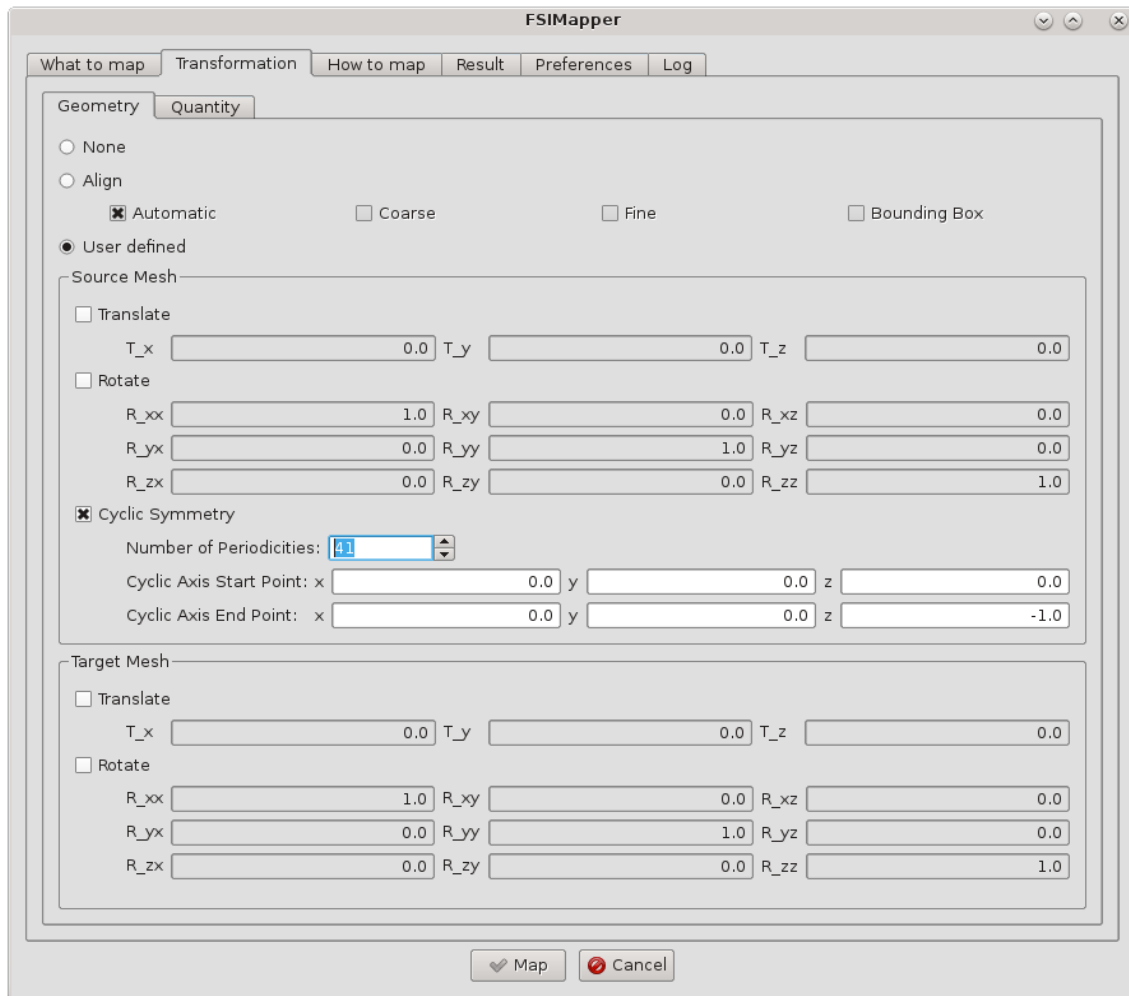Figure 44: The "What to map" panel
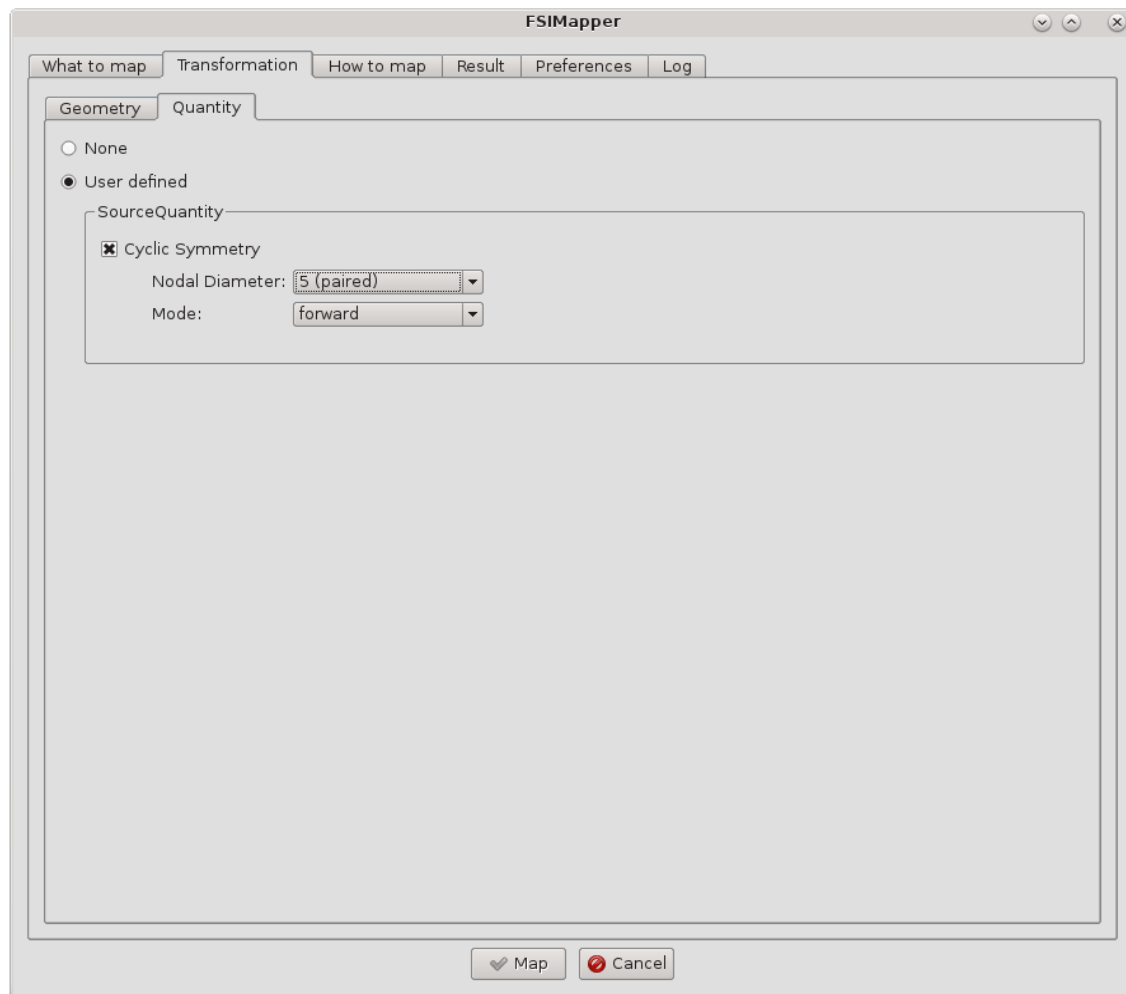
Figure 45: The "Transformation/Geometry" panel

Figure 46: The "Transformation/Quantity" panel

When the mapping finishes, the MpCCI Visualizer shows the mapping results, cf. Figure 47.

(a) Full source model

(b) Mapping result (imaginary pressure) on target model with superimposed source model section shape (black outlines)
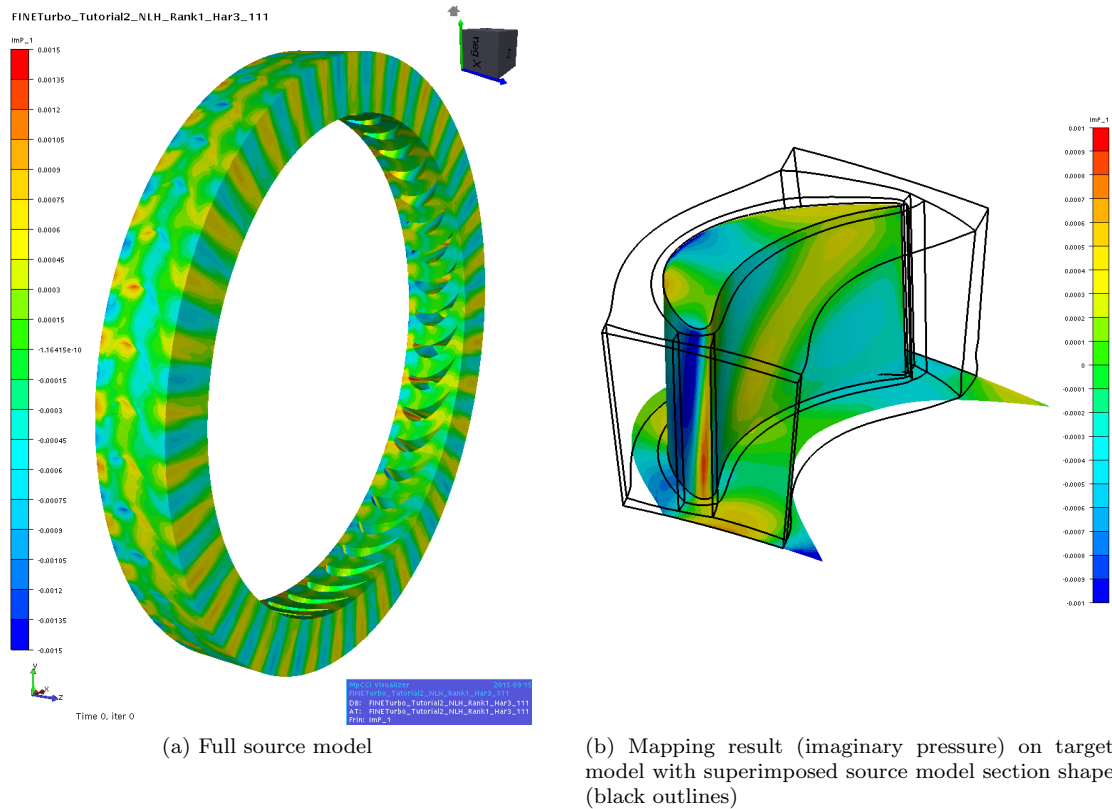
Figure 47: Harmonic mapping result shown in MpCCI Visualizer

The periodic mesh and data were transformed by the given information in order to create the full source model and the corresponding data (cf. Figure 47a). The first harmonic complex pressure excitation has been revolved such that the data are spacially continuous over the periodic boundaries, which ensures a correct mapping to the target surfaces which are not covered by the periodic source mesh, see Figure 47b. The continuity over the periodic source boundaries is only provided for the correctly selected nodal diameter and excitation mode.

MpCCI FSIMapper creates two Abaqus include files "aachenBlade-mapped_HarmonicPressure_RE.inc" and
"aachenBlade-mapped_HarmonicPressure_IM.inc" where the real and imaginary part of the pressure is applied to the nodes building the wetted surfaces:

```
**
** Exported real part of surface pressure
** *CLOAD, REAL
37881,  8,  0.00040665
37882,  8,  0.00046127
37883,  8,  0.00038815
37884,  8,  0.00032466
37885,  8,  0.00037576
37886,  8,  0.00031139
...
```

```
**
** Exported imaginary part of surface pressure
** *CLOAD, IMAGINARY
37881,  8,  -0.00018233
37882,  8,  -0.00010771
37883,  8,   0.00001771
37884,  8,  -0.00033982
37885,  8,  -0.00023237
37886,  8,  -0.00014165
...
```

(!) For an element-based pressure definition, select "Element" in the "How to map" panel in "Quantity location (target)". The keyword defining the pressure is then `*DLOAD` instead of `*CLOAD`.

Moreover, MpCCI FSIMapper creates the file `"aachenBlade-mapped_HarmonicPressure.ccvx"`. It can be opened in MpCCI Visualizer and shows the corresponding transient pressure fluctuations for 18 pseudo time steps. Pressing the ▶ button animates them with a speed set in File→Preferences→Maximum animation FPS.

8.2.2.5 Target Simulation

MpCCI FSIMapper exported for the first harmonic two include files containing the real and the imaginary part of the pressure excitation. They are located in the same folder as the Abaqus input deck. The load case definition in Abaqus uses these files in the following way:

```
**
*CYCLIC SYMMETRY MODEL, N=41
0, 0, 0, 0, 0, -1
**
...
**
*STEP, NAME=ModalAnalysis
**
*FREQUENCY, EIGENSOLVER=LANCZOS
 , 0, 5250, , , ,
*SELECT CYCLIC SYMMETRY MODES, NMIN=5, NMAX=5
**
*END STEP
**
**
**
*STEP, NAME=SSD_1H
**
*STEADY STATE DYNAMICS
2100, 0, , ,
**
*CLOAD, REAL, CYCLIC MODE=5
*INCLUDE, INPUT=aachenBlade-mapped_HarmonicPressure_RE.inc
**
*CLOAD, IMAGINARY, CYCLIC MODE=5
*INCLUDE, INPUT=aachenBlade-mapped_HarmonicPressure_IM.inc
**
```

```
*END STEP
**
```

The periodicity of the "Aachen" rotor blade is defined in Abaqus via the keyword `*CYCLIC SYMMETRY MODEL`. Since the excitation is in forward mode with respect to the negative z-axis, `0, 0, 0, 0, 0, -1` is given here to define the symmetry axis. The first harmonic pressure will excite only nodal diameter 5 mode shapes, which are determined in the `*FREQUENCY` step.
The excitation shape needs to be defined in the keyword `*CLOAD` by the option `CYCLIC MODE=5`.

Please check the Abaqus User's Keywords Reference Guide for the options of `*STEADY STATE DYNAMICS` concerning direct and modal solution methods.

(!) For the sake of completeness use also the mean pressure (by a "Static" mapping), the centrifugal and the Coriolis forces (by the keyword `*DLOAD`).

(!) The damping in the system has fundamental influence to the magnitude of blade vibration, so it is recommended to define it.

(!) Since Abaqus only knows forward excitation modes, for backward excitations the cyclic symmetry axis (which was determined such that the rotation is positive) has to be turned around (only) for the definition in the Abaqus keyword `*CYCLIC SYMMETRY MODEL`.

8.3 Mapping of Temperature for Microelectronic Devices

8.3.1 Problem Description

In this tutorial, called "Board", the stationary temperature field inside a microelectronic device was simulated using Mentor Graphics computational fluid dynamics software FloTHERM. By a volume mapping of the resulting temperature field to a structural mechanics model a thermal stress analysis can be performed. Here, the mapping of temperature field is demonstrated for the software packages Abaqus, ANSYS and MSC Nastran.
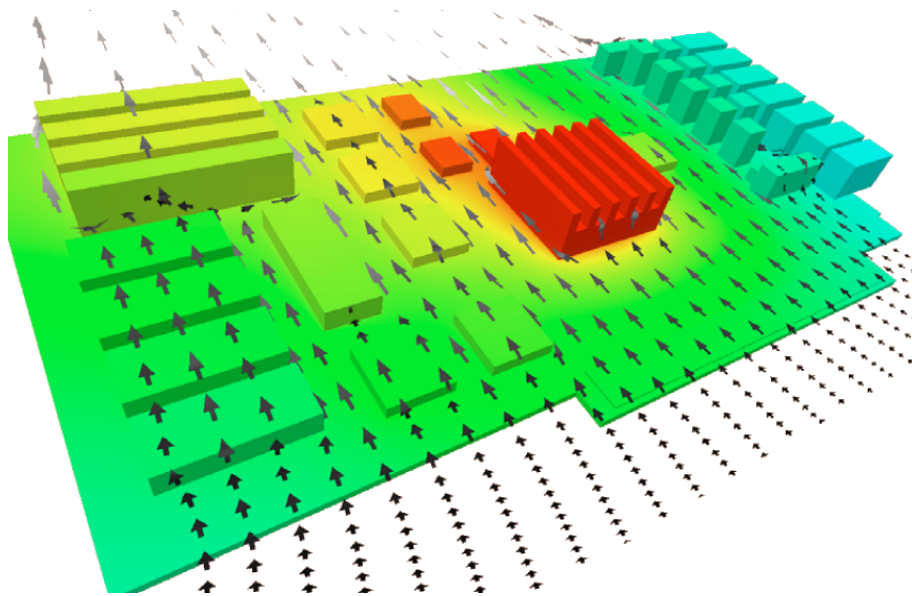

Figure 48: Temperature destribution and external flow field simulated by FloTHERM

The files concerning this tutorial are located in

- MpCCI FSIMapper as part of the MpCCI installation:
  "<*MpCCI_home*>/tutorial/FSIMapper/Board"

- MpCCI FSIMapper standalone installation:
  "<*MpCCIFSIMapper_home*>/tutorial/Board"

Before mapping, copy the files to your working directory.

8.3.2 Source Result File

The stationary simulation of the temperature field under operating conditions was performed using FloTHERM. The native FloTHERM model is not part of the MpCCI FSIMapper tutorial but its result exported (as .flofea file) from FloTHERM project manager. To export a simulation result from FloTHERM V10 and later, select the assembly in the Model panel. Using right-click on selected assembly, choose format "FLOFEA" below "Export Assembly".

The FloTHERM model consists of 14611 hexahedron elements, as length unit meter and temperature unit Celsius is used.

8.3.3 Target Mesh Files

The structural target mesh consists of 1279 8-node brick elements, as unit system mm-ton-second-celsius is used.

Abaqus

In Abaqus DC3D8 elements are used. The model is split up in three Abaqus parts. Each part has a single part instance in the Abaqus *ASSEMBLY section. To receive volumetric temperature distribution, one *ELSET definition per part is created.

ANSYS

In ANSYS SOLID185 elements are used. The elements of the model are grouped in three components using the ANSYS cm command.

MSC Nastran

In MSC Nastran CHEXA elements are used. The elements of the model are assigned to three part ids.

8.3.4 Mapping of Temperature

Open the MpCCI FSIMapper GUI and go to the "Static" tab of the "What to map" panel.

1. As source code select FloTHERM in the format selection dropdown menu.
2. Click on "Specify source geometry file" and select `"Board/FloTHERM/board.flofea"`
3. Confirm selection by pressing Open
4. The selected file is scanned in background and available parts for mapping are listed
5. Select all parts of the source model
6. In "Temperature" dropdown menu select "TEMPERATURE"
7. As target code select:

Abaqus

   a) Abaqus in the format selection dropdown menu.
   b) Click on "Specify target geometry file" and select `"Board/Abaqus/board.inp"`
   c) Confirm selection by pressing Open
   d) The selected file is scanned in background and available parts for mapping are listed
   e) Select Elset -1 "Chips_All_Elements", Elset -2 "Board_All_Elements" and Elset -3 "Cooling_All_Elements"
   f) The "What to map" panel of the mapping process now should look as shown in figure 49

ANSYS

   a) ANSYS in the format selection dropdown menu.
   b) Click on "Specify target geometry file" and select `"Board/ANSYS/board.db"`
   c) Confirm selection by pressing Open
   (!) Scanning of ANSYS.db files requires a present ANSYS Mechanical APDL installation If no installation is available for this tutorial select `"Board/ANSYS/board.ml"` instead. See section 5.2.4 for details about scan process.
   d) The selected file is scanned in background and available components for mapping are listed
   e) Select Volume 1 "BOARD", Volume 2 "CHIPS" and Volume 3 "COOLING"

f) The "What to map" panel of the mapping process now should look as shown in figure 49

MSC Nastran

a) MSC Nastran in the format selection dropdown menu.
b) Click on "Specify target geometry file" and select `"Board/MSC.Nastran/board.bdf"`
c) Confirm selection by pressing Open
d) The selected file is scanned in background and available parts for mapping are listed
e) Select Volume 1 "1", Volume 2 "2" and Volume 3 "2"
f) The "What to map" panel of the mapping process now should look as shown in figure 49

8. Go to the "How to map" panel and select Node in the "Quantity location" section

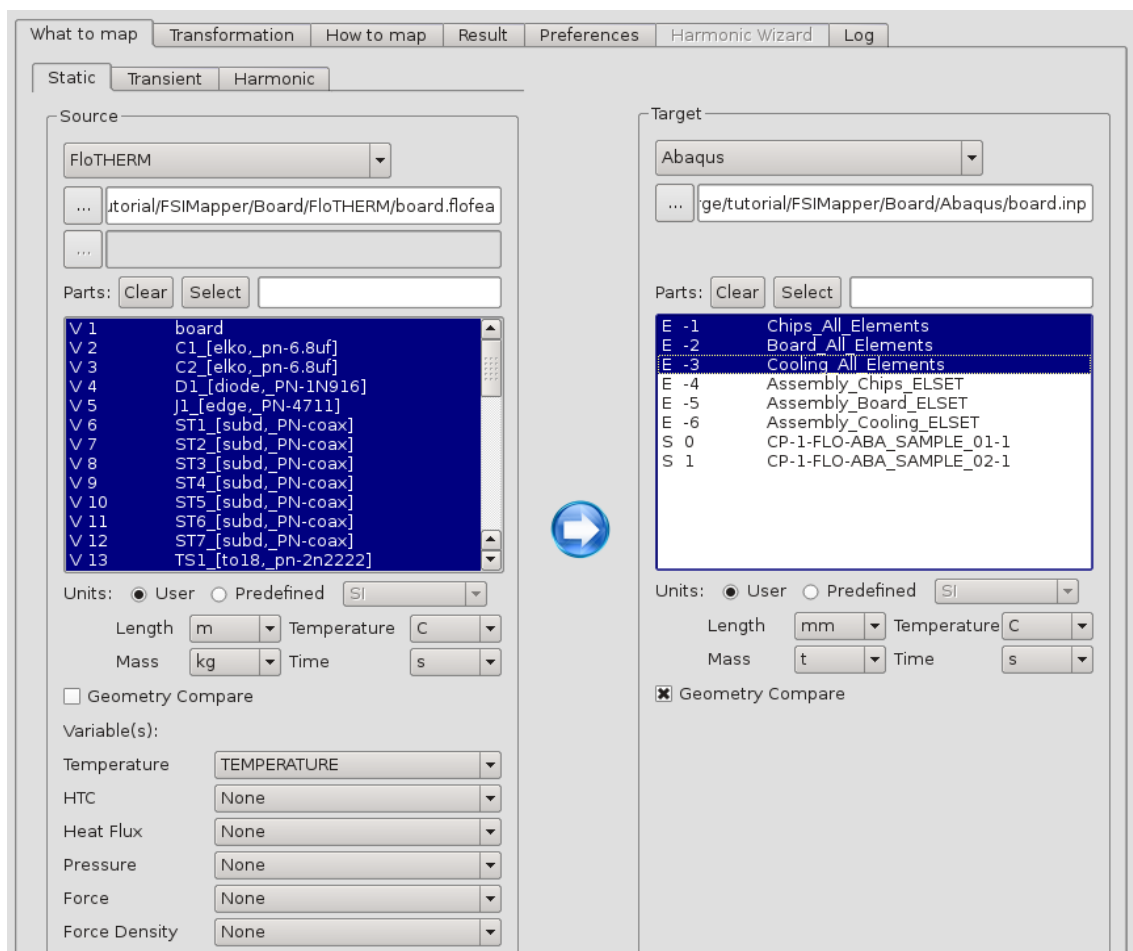9. Press Map at the bottom of the MpCCI FSIMapper GUI and perform mapping

Figure 49: The "What to map" panel after step 7 for target code Abaqus

8.3.5 Results and Target Simulation

After pressing the "Map" button, using the predescribed process setup, the MpCCI FSIMapper runs the mapping. In the background both source and target model as well as the temperature distribution are
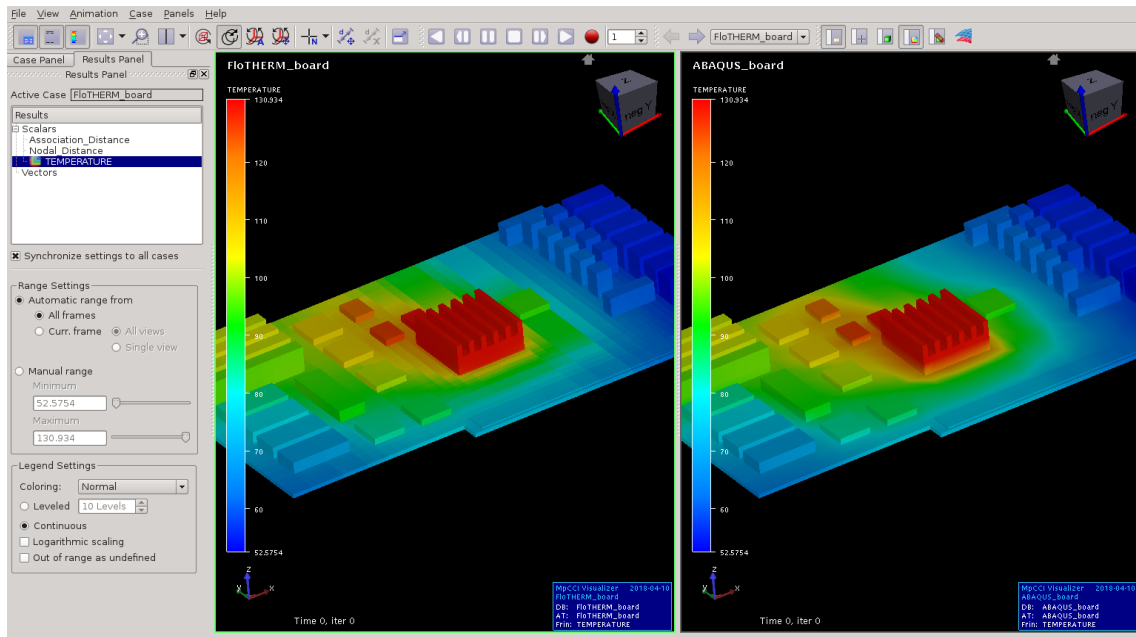
Figure 50: Temperature mapping result of the microelectronic device at the end of the mapping process.

read from file system and the data is interpolated to the new geometry. Then, all data is sent to MpCCI Visualizer for postprocessing (cf. figure 50). Then, MpCCI FSIMapper exports nodal based temperature to an ASCII-based input file in the native solver syntax. They are located in the same folder as the structural mechanics input deck.

Abaqus

For Abaqus the include file `"board-mapped_Temperature.inp"` is written, wherein the `*TEMPERATURE` keyword is used to define initial temperature values.

```
** Exported nodal temperature
** *TEMPERATURE
ChipsInstance.1,95.07985578
ChipsInstance.2,87.69397911
ChipsInstance.3,83.75685351
ChipsInstance.4,91.68932411
ChipsInstance.5,91.48773958
ChipsInstance.6,91.26212328
```

The temperature mapping result file can be included in another Abaqus simulation using the following include statement at `*STEP` level of the loadcase definition:

```
*INCLUDE, INPUT=board-mapped_Temperature.inp
```

ANSYS

For ANSYS the include file `"board-mapped_Temperature.inc"` is written, wherein the BF keyword in compination with TEMP is used to define initial temperature values.

```
C*** Exported nodal temperature
/PREP7
BF,        1,TEMP, 9.507985421E+01
BF,        2,TEMP, 8.769398132E+01
BF,        3,TEMP, 8.375684495E+01
BF,        4,TEMP, 9.168932019E+01
BF,        5,TEMP, 9.148773962E+01
BF,        6,TEMP, 9.126212312E+01
```

The temperature mapping result file can be included in another ANSYS simulation by "Read Input" inside
ANSYS Mechanical APDL.

MSC Nastran

For MSC Nastran the include file `"board-mapped_Temperature_SPC.inc"` is written, wherein the SPC
keyword is used to define initial temperature values.

```
$ Exported nodal temperature
SPC         1        1        195.07985
SPC         1        2        187.69398
SPC         1        3        183.75684
SPC         1        4        191.68932
SPC         1        5        191.48774
SPC         1        6        191.26212
```

The temperature mapping result file can be included in another MSC Nastran simulation using the following
include statement:

```
INCLUDE 'board-mapped_Temperature_SPC.inc'
```