

MpCCI
Mapper

Version 4.5.2

<http://www.mpcci.de/en/mpcci-software/mapper.html>



Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Abaqus and **SIMULIA** are trademarks or registered trademarks of Dassault Systèmes

ANSYS is a registered trademarks of Ansys, Inc.

ARGUS, **ARAMIS** and **ATOS** are registered trademark of Gesellschaft für Optische Messtechnik mbH

AutoGrid is a registered trademark of ViALUX Messtechnik + Bildverarbeitung GmbH

AutoForm is a registered trademark of AutoForm Engineering GmbH

Cadmould is a registered trademark of SIMCON Kunststofftechnische Software GmbH

Forge is a registered trademark of TRANSVALOR S.A.

Indeed is a trademark of Gesellschaft für Numerische Simulation mbH

LSDyna is a registered trademark of Livermore Software Technology Corporation

Moldflow is a registered trademarks of MF SOFTWARE GmbH

MSC Nastran, **MSC Marc** and **MSC SIMUFACT** are registered trademarks of MSC.Software Corporation

COPRA FEA is a registered trademarks of data M Sheet Metal Solutions GmbH

PAMStamp, **PAMCrash** and **SYSWELD** are trademarks or registered trademarks of ESI Group

RADIOSS is a registered trademark of Altair Engineering

SIMUFACT is a registered trademark of simufact engineering gmbh

Linux is a registered trademark of Linus Torvalds

Windows, **Windows XP**, **Windows Vista**, **Windows 7**, **Windows 8**, **Windows 8.1** and **Windows 10** are registered trademarks of Microsoft Corp.

Contents

1	Release Notes	4
1.1	Version 4.5.2	4
1.2	Version 4.5.1	5
2	Installation	6
2.1	Tool	6
2.2	FlexLM	6
3	Using the MpCCI Mapper	10
3.1	Menus	10
3.2	Toolbar	14
3.3	Viewport	14
3.4	Data Panel	16
3.5	Postprocessing	22
3.6	Mapping	25
3.7	Batch Mode	26
4	Solver Formats	31
4.1	Abaqus	31
4.2	ANSYS	33
4.3	AutoForm R4 converter	34
4.4	Cadmould	35
4.5	Indeed	35
4.6	Forge / SIMUFACT	36
4.7	LS Dyna	37
4.8	Marc / COPRA FEA	41
4.9	Moldflow	42
4.10	Nastran	42
4.11	PAM	43
4.12	Radioss	43
4.13	Sysweld	44
5	Other Formats	45
5.1	Argus / Aramis	45
5.2	Atos	45
5.3	AutoGrid	45
5.4	STL	45
5.5	VTK	46
6	Thirdparty License Information	47
6.1	LibQxt	47
6.2	Libtiff	47
6.3	Libwebp	48
6.4	LibXBC	48
6.5	PCRE Library	49
6.6	VTK	50

1 Release Notes

1.1 Version 4.5.2

List of changes and improvements since MpCCI Mapper version 4.5.1

1. Graphical user interfaces has been redesigned and is now QT based 3
2. Validation concept has been redesigned and is now available for all quantities 3.4.2
3. Enhanced scriptable batch mode using configuration file 3.7.5
4. Enhanced data visualization for tensor and vector quantities 3.4.2

Table 1.1: GUI related changes and extensions

5. Abaqus export of shell section composite when mapping fibre orientations 4.1
6. ANSYS export support for THICKNESS variable 4.2
7. COPRA FEA reading support (based on MSC Marc) 4.8
8. LSDyna history variable concept has been extended to assign higher history ids than defined
9. MSC Marc .t19 result file reading support 4.8
10. Moldflow can handle multiple XML-based result files at once 4.9
11. MSC Nastran SOL 700 write support for stresses and strains 4.10

Table 1.2: Solver formats changes and extension

1.2 Version 4.5.1

List of changes and improvements since MpCCI Mapper version 4.5.0

1. Additional MpCCI Mapper arguments for batch mode usage 3.7
2. Inverse alignment direction if target model is larger than the source model 3.1.2
3. Quantity renaming support in quantity dialog 3.4.2
4. Slice and length plot does support multiple parts; fixed plane widget interaction 3.5.2

Table 1.3: GUI related changes and extensions

5. Solver Abaqus 4.1
 - Write support for *ORIENTATION keyword defining local material directions for anisotropic materials
6. Solver Cadmould result reading support 4.4
7. Solver LSDyna 4.7
 - ORIENTATION_TENSOR from injection molding can be used to create *ELEMENT_SOLID_ORTHO card with material ids depending on degree of orientation
 - Mapping support of material orientation from MAT249 to MAT58
 - *PART_COMPOSITE card reading support
 - Scaling of History variables on data export
8. Solver Moldflow result reading support 4.9

Table 1.4: Solver formats changes and extension

2 Installation

2.1 Tool

MpCCI Mapper comes along as packed archive containing a platform specific executable 'mapper.exe' which can be unpacked to an arbitrary working directory and is ready to use after setting up necessary license information (c.f. section 2.2.2 for detailed description). On samba mounted directories take care of the execute flags to be set for the mapper.exe.

2.2 FlexLM

MpCCI Mapper uses a FlexLM based floating license mechanism. FlexLM license server has to be started on the license server host defined in the MpCCI Mapper license file. You can run MpCCI Mapper anywhere in your internal network. For more information about FlexLM, please refer to the FlexLM end-user's guide which is normally located in `<MAPPER_HOME>/license/LicenseAdministration.pdf`).

2.2.1 Prerequisites

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

With MpCCI Mapper all FlexLM platform dependent binary executables are installed in a separate directory (`<MAPPER_HOME>/license/<architecture>`). Your license file should have the name 'mpcci_SVD_Your_Company_Expiration_Date.lic'.

MpCCI Mapper only needs three FlexLM executables:

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

The MpCCI Mapper vendor daemon is named SVD. The FlexLM port number of the SVD vendor daemon is 47000 by default. If there are other software packages installed using also FlexLM there will be several FlexLM 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.

2.2.2 Installing License

After installation of MpCCI Mapper you need to acquire a license file from Fraunhofer SCAI.

Please login on the host where the FlexLM license server for MpCCI Mapper should run on. In the following example, "\$" is your prompt:

```
$ hostname  
myHostName
```

```
$ lmutil lmhostid -n  
12345abcd
```

Please send hostname (in the above example: "myHostName") and hostid to "mpcci@scai.fraunhofer.de".

If you have multiple network devices installed (Ethernet card, Wireless LAN, Docking Station on a Notebook) you may see multiple hostids. For the MpCCI Mapper license daemon the integrated ethernet card address should be the correct one.

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

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

To stop the license server please type:

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

You should set the generic FlexLM variable SVD_LICENSE_FILE.

For ksh or bash user:

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

For csh users:

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

This variable is also referred to by other software packages using FlexLM; LM_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 Unix/Linux and by a ";" on MS Windows.

We suggest that you set the SVD_LICENSE_FILE environment variable in your login file (.cshrc or .profile or .bashrc). FlexLM stores data of a previous successful connection to a license server in the file "<HOME>/flexlmrc". You may create or adjust this file "<HOME>/flexlmrc".

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 Mapper :

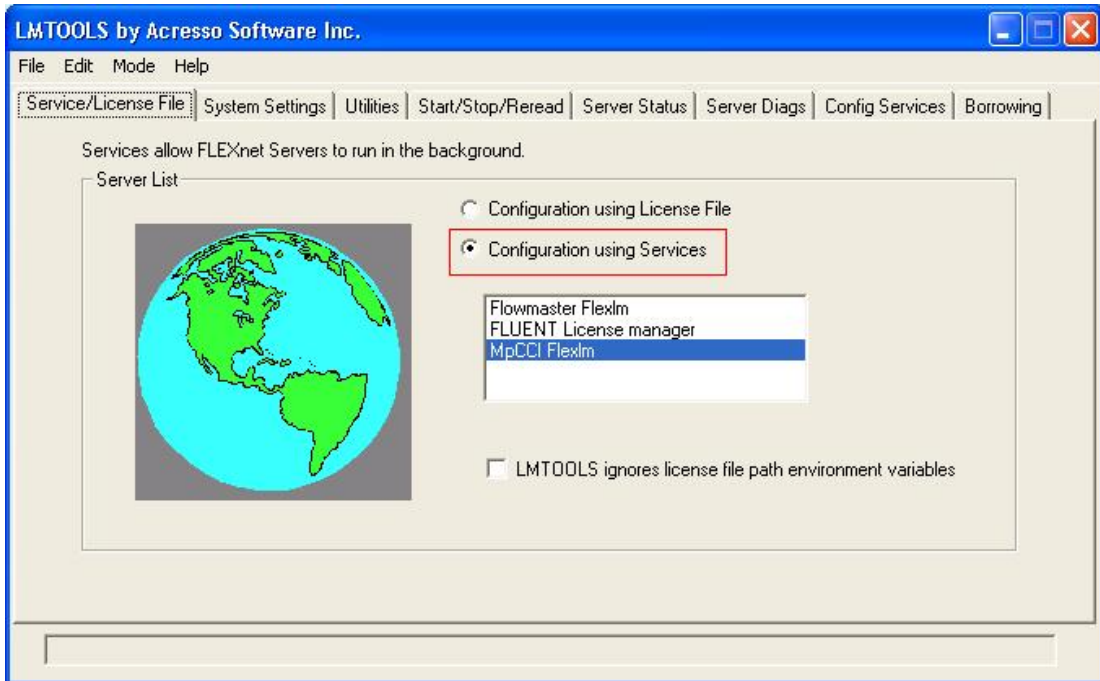
```
$ lmutil lmstat -vendor SVD
```

For more details

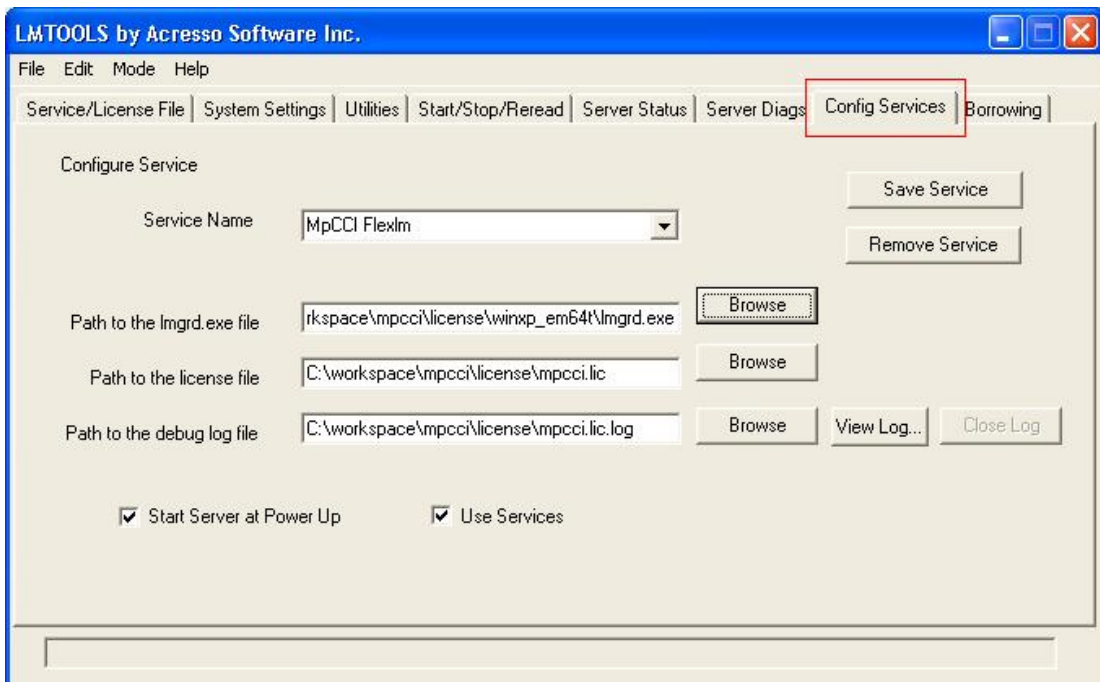
```
$ lmutil lmstat -a -c <LICENSE_FILE>
```

2.2.3 Configure a License Manager as Windows service

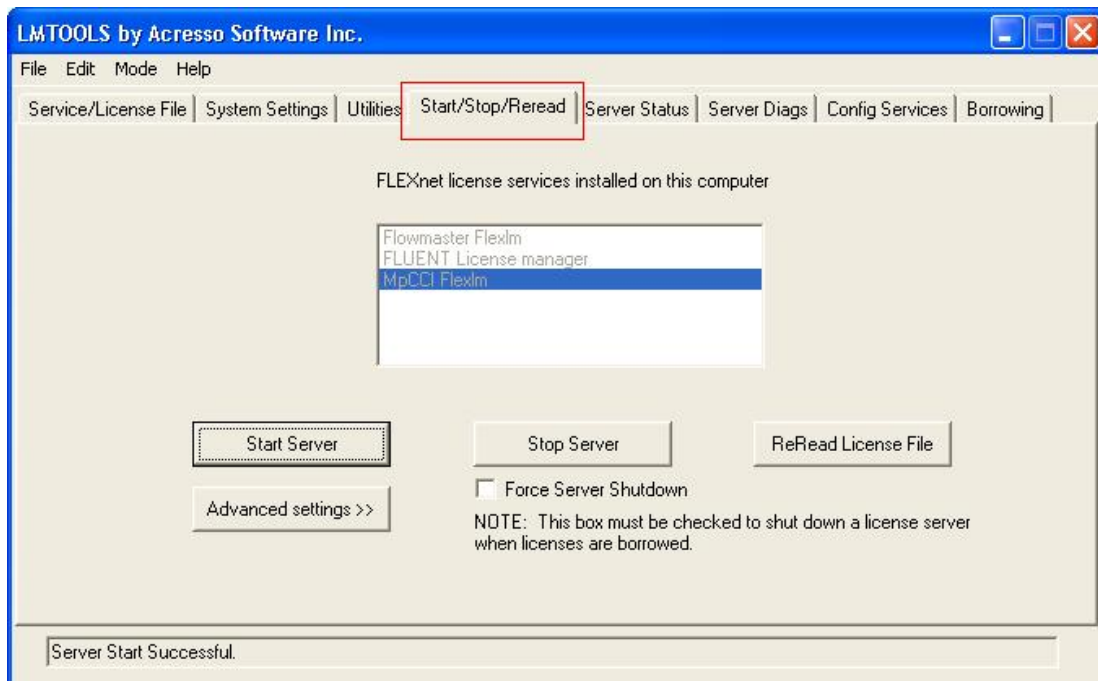
Execute the *lmtools.exe* application from the license manager installation directory:
< *LICENSE_TOOL_INSTALLATION* > /bin/*lmtools.exe*



- 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 Using the MpCCI Mapper

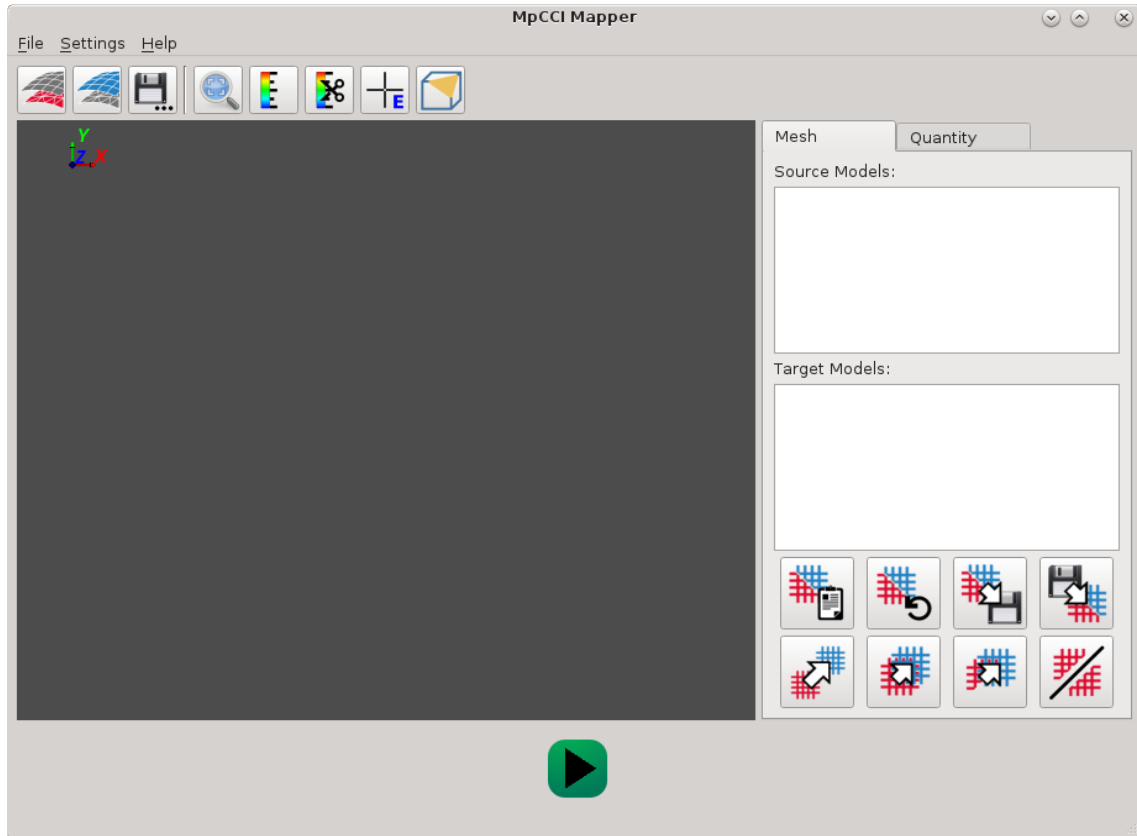


Figure 3.1: Graphical user interface of MpCCI Mapper

The graphical user interface of the MpCCI Mapper is split up in three major areas. Located on top of the application, a list of action buttons form the so called 'Toolbar' 3.2 where basic operations of the MpCCI Mapper can be accessed. Right below the toolbar, the main viewport 3.3 handles the display of model geometry and quantities in a OpenGL rendering area. Alongside of the viewport, the data panel 3.4 is divided in a mesh and a quantity related subpanel to setup a mapping step. Geometry related information, e.g. models, are shown in the 'Mesh' panel 3.4.1 where opened source and target meshes with their parts are listed. Quantity related information, e.g. the definition of mapping quantities or the validation of a quantity, can be done in the 'Quantity' panel 3.4.2. At the bottom of the window a single button is present which is responsible for starting mapping or validation.

3.1 Menus

3.1.1 File Menu

The file menu located at the top left corner of the graphical user interface 3.1. On left click on it the so called 'file menu' opens as displayed in figure 3.2 offering several options

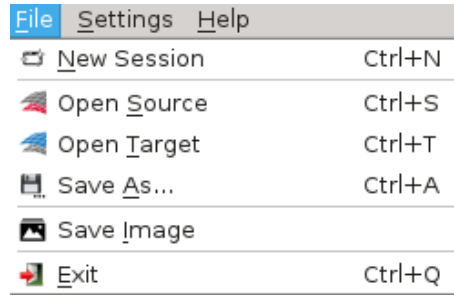


Figure 3.2: File Menu Content

for interaction with local file system. A detailed description on each option is given in table 3.1.

Entry	Shortcut	Description
New Session	Ctrl+N	Delete all currently loaded models and reset program
Open Source	Ctrl+S	Open a source model (often the metal forming results)
Open Target	Ctrl+T	Open a target model to map to (often the crash model)
Save As	Ctrl+A	Save a target result file. Some file formats support patching which means that the original file is read in, the new mapped values are added and a merged file is written.
Save Image	Alt+F, I	A picture from the visualisation window is captured and stored in a file. The resolution of the image is quite higher than that of the screen. Take care if you save bitmaps (size > 30Mbyte)
Exit	Ctrl+Q	Closes the application

Table 3.1: Options of File Menu

3.1.2 Settings Menu

The settings menu is located at the top menu bar of the graphical user interface next to the file menu 3.3. It is designed as a drop down menu containing certain alignment, mapping algorithm and file writer configuration options.

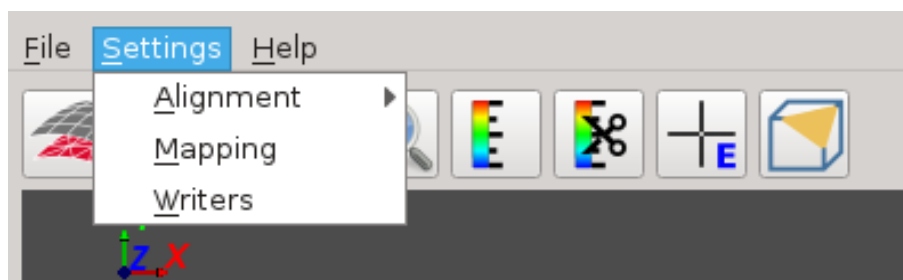


Figure 3.3: Settings Menu

Alignment Settings

The alignment direction of the MpCCI Mapper can be inverted in "Settings/Alignment". If target model is larger than the mapping source model, automatic alignment 3.4.1 tools

might produce poor alignment quality. Using the inverse alignment direction, the alignment quality can be significantly improved.

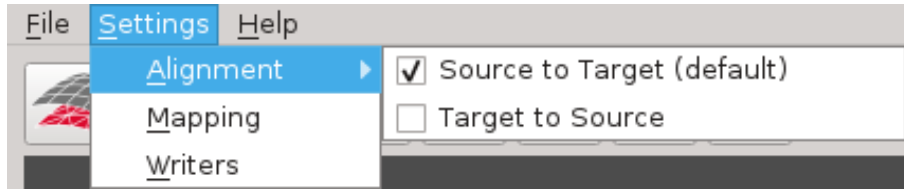


Figure 3.4: Inverse alignment direction from default source on target to target on source.

Mapping Settings

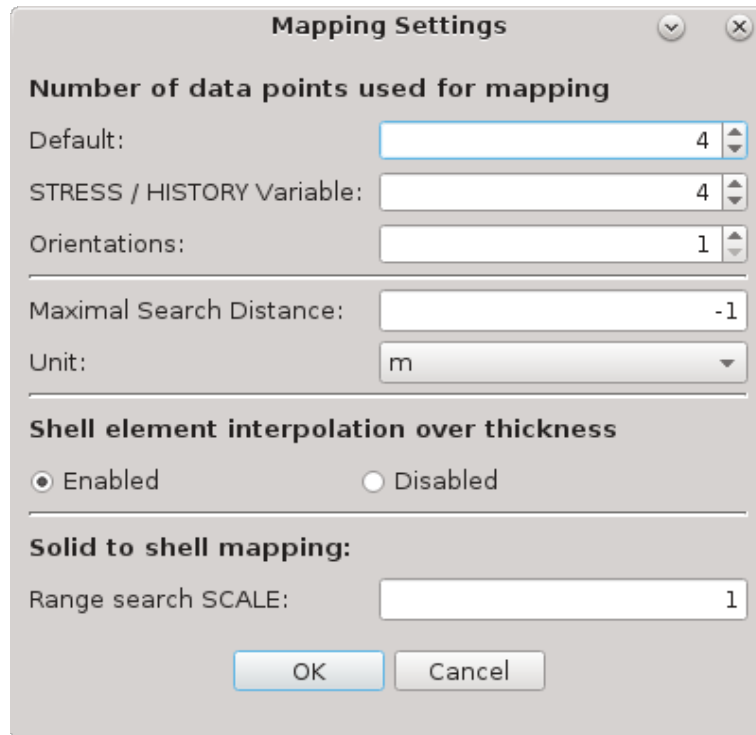


Figure 3.5: Mapping settings

MpCCI Mapper uses a k-nearest-node mapping algorithm (default $k = 4$) that combines robustness with data smoothing. At times where averaging of certain entities is not wanted or when data smoothing shall be increased MpCCI Mapper allows to adjust the number of data points used for interpolation. This can be done in "Settings/Mapping" as shown in figure 3.5 where parameter k (valid range between $k=1$ and $k=8$) can be set for explicitly quantity STRESS (and HISTORY variables for LSDyna), Material ORIENTATIONS or DEFAULT for all quantities.

Having source and target models with different number of integration points over shell element thickness, an interpolation between the thickness integration is done. If a user does not want interpolation over element thickness, it can be disabled so the interpolated value is determined by the nearest integration type.

If the source model is a subset of the target model, and data shall only be mapped within source model range, a maximal search distance can be defined.

To perform a solid to shell mapping MpCCI Mapper uses a surface to surface projection approach to determine THICKNESS information from solid geometry. As an initial step the upper and lower surfaces of the shell model get translated by half of initial shell thickness in element normal direction. In some cases (shell surface inside solid) it might be necessary to increase the shell surface translation by an addition factor. The factor “SCALE” (default = 1) can be defined in the corresponding line edit entry as shown in figure 3.5.

Writer Settings

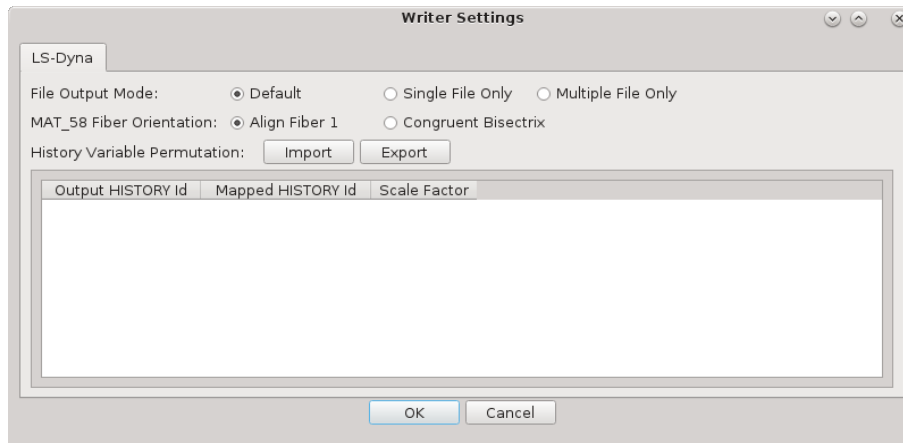


Figure 3.6: Writer settings

Using the “Settings/Writer” menu entry it is possible to configure the LSDyna file writer. At first the writer can be configured to export single or multiple files. In addition, an exchange of the position and a data scaling of history variables can be defined during export. Section 4.7.6 gives an overview about how to define the ASCII-based configuration file.

3.1.3 Help Menu

The help menu gives information about MpCCI Mapper version and has an easy access to this documentation by pressing F1.

3.2 Toolbar

The MpCCI Mapper toolbar is located at the top of the graphical user interface. It offers functionality for view navigation as well as shortcut buttons to file import and export of the file dialog. Figure 3.7 gives an overview on possible actions.









	Load a new source model from file
	Load a new target model from file
	Save active (visible) target models
	Reset Camera to show all active models
	Adjust the quantity color range by the user or reset to automatic range
	Clip a scalar value range Values above and below new limits are set to lower resp. upper limit
	Show info about selected element 3.5.1
	Slice through model and plot scalars along traverse 3.5.2

Figure 3.7: Toolbar button description

3.3 Viewport

The viewport of the graphical user interface is the area where loaded models are displayed. Herein the user can interactively inspect an arbitrary number of models at the same time. Descriptions for use of the viewport by mouse and hotkeys are given in sections 3.3.1 and 3.3.2;

3.3.1 Basic View Interaction

Spinning of models

Press and hold left mouse button on a model

Moving model left/right up/down

Press and hold middle mouse button on a model

Zooming models

Press and hold right mouse button on a model

Spinning camera

Press and hold left mouse button on background

Moving camera left/right up/down

Press and hold middle mouse button on background

Moving camera forward, backward

Press and hold right mouse button on background

Table 3.2: Mouse interaction and navigation

3.3.2 Viewport Hotkeys

'w'	Wireframe mode: active parts are presented as wireframes
's'	Surface mode: active parts are presented as surfaces
'b'	active parts are presented as surfaces with element discretization overlay
'p'	Points mode: active parts are presented as point clouds
'o'	Opaque mode: toggle opacity/transparency
'e'	toggle element information mode on/off
'i'	toggle cut plane widget on/off

Table 3.3: List of available hotkeys for viewport buttons

Remark: To change the style of only some parts deactivate the others, switch the style and activate the others again.

3.4 Data Panel

The data panel organizes geometry and quantity related information. It is divided in a 'Mesh' section 3.4.1 where opened source and target meshes with their parts are listed and a 'Quantity' section 3.4.2 where quantities for mapping and validation are managed.

3.4.1 Mesh Panel

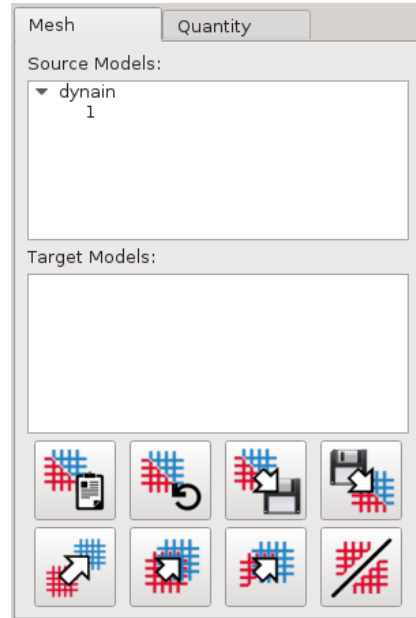


Figure 3.8: Mesh Panel

The 'Mesh' panel can be used to manage models as well as their relative position to each other. Here, each loaded model is assigned to the list of source or list of target models. For each model, a so call model tree gives an overview about the model and its subset partitions. Single parts or models can be activated or deactivated for visualization and an unit system can be assigned for each model separately.

At the bottom of the 'Mesh' panel, a form based layout offers different automatic model alignment options 3.4.1 which can be used to place models before data exchange.

Model and Part Selection

Double clicking on model name

Models are activated or deactivated (for mapping, aligning and visualization)

Double clicking on part name

Parts are activated or deactivated (for mapping, aligning and visualization)

Click on triangle

Collapse or expand the parts tree

Right click on model name

Pop up of Models Context Menu

Table 3.4: Part selection options

Models Context Menu

On right click on model tree the 'models context menu' opens.

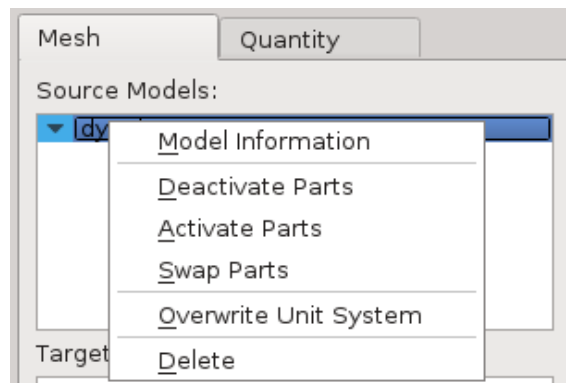


Figure 3.9: Models Menu

Model Information

Shows Information about the mesh topology and quantities

Deactivate Parts

Deactivates all parts of model (for mapping, aligning and visualization)

Activate Parts

Activate all parts of model (for mapping, aligning and visualization)

Swap Parts

Deactivates all activated and activates all deactivated parts

Overwrite Unit System

During reading all data (point coordinates, thickness, stress..) is transformed to SI-units. During writing all data is transformed back to the original unit system. Use this option if the first guess of the unit system was wrong.

Delete

Removes mesh from the GUI and database

Automatic Positioning



Figure 3.10: Automatic model alignment tools

A common problem in data mapping between different simulation models is the use of varying local coordinate systems. Here, the models might not have the same spatical location in a common coordinate system. Hence, for a position-dependent neighborhood

computation, both models need to be aligned in a preprocessing step. This alignment step can be done using the automatic positioning features in MpCCI Mapper (c.f. table 3.11).









	Remember current position for mapping and validation
	Set model positions to last remembered position.
	Rough Align: Used to bring equal meshes close to each other.
	Fine Align: Align meshes for mapping. Can be used several times to improve the result.
	Symmetric Align: Used if one of the models uses axes symmetry
	Mirror Geometry: Mirrors all active models Warning: Mirroring is applied to the geometry. When saving the mapping results of a mirrored target, the mirrored geometry is written to the output file.
	Save current positions to a file: 1. It is possible to store the transformations in Mapper format (*.trf) 2. Export target mesh nodes in current position in LSDyna keyword format 3. Export source mesh nodes in current position in LSDyna keyword format
	Load transformation from a file: It is possible to read the transformation of a target to a source model in MpCCI Mapper format. Transformation needs to be initialized with keyword ”*transformation”. Declation of the transformation begins with a new line and is read linewise; entries need to be separated by blanks. Note: All active targets will be transformed.

Figure 3.11: List of available automatic positioning methods.

If the transformation of both coordinate systems are know, a manual setup of the MpCCI Mapper transformation file can be done in he following way:

Example Transformation

If $A = (a_{ij})_{i,j=1,\dots,3}$ is the rotation matrix of a model in the 3-dimensional space and $T = (t_i)_{i=1,\dots,3}$ the translation vector we have the transformation Q in homogenous coordinates

$$Q = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
*transformation = -3.289344e-05  -1.000000e-00  1.079831e-06  2.674954e-04
                  5.307190e-04  1.062373e-06  9.999999e-01 -3.006939e-01
                  -9.999999e-01  3.289400e-05  5.307189e-04  1.028435e+00
                  0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00
```

3.4.2 Quantity Panel

The 'Quantity' panel, as second entry of the data panel, lists up quantities which are currently loaded inside MpCCI Mapper and are available for mapping to other models or for validation purpose.

Mapping Quantity Selection

In the 'Mapping' section (c.f. figure 3.12) of the quantity panel a list of all available quantities is shown. By default, all present quantities are set active for mapping. If only a subset should be used, unwanted transfer quantities can be deactivated in the list overview by direct user interaction 3.4.2 or semi-automatic using the quantity context menu 3.4.2.

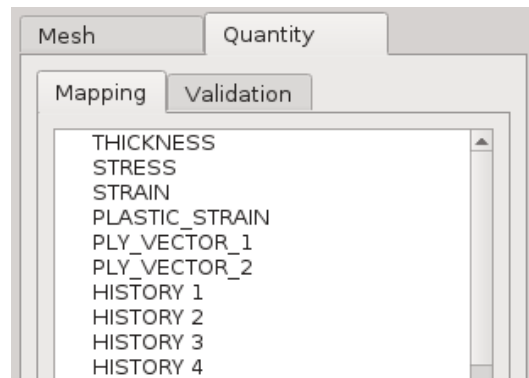


Figure 3.12: Selection of quantities for mapping

Left Click

Select/deselect quantity for visualization of values as contour plot

Double (Left or Mid) Click

Activate/deactivate quantity for mapping (black/grey letters)

Right Click

Open quantity dialog box

Deselect

Show parts

Table 3.5: Mapping quantity interaction

Remark:

Averaged values of all element integration points are used for element based data visualization. This may lead to varying color display if source and target have different number of integration points.

Quantity Context Menu

By right clicking on a quantity name the quantity context menu opens. Here, automatic selection and deselection operations on quantities can be done or additional information on especially tensor data can be requested for display (c.f. table 3.4.2).

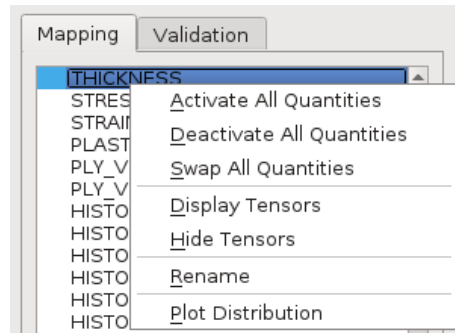


Figure 3.13: Overview of the Quantity Context Menu

Activate All Quantities

Activates all defined quantities for mapping

Deactivate All Quantities

Deactivates all quantities

Swap All Quantities

Activates currently deactivated quantities, deactivates currently activated quantities

Display Tensors

Show principal axes of a tensor quantity

Hide Tensors

Hides principal axes of a tensor quantity

Rename

Assign a new name for selected quantity

Plot Distribution

Shows a percentual distribution of selected quantity per mesh in a 2D plot

Table 3.6: Quantity Dialog Options

Quantity Validation

The Validation panel is designed for validating mapped and non-mapped quantity data on given source and target meshes. By default all quantities defined on source and target model will be used during validation process. For individual configuration of quantities use the right click context menu 3.15.

Note: Validation functionality can also be used to compare nonmapped quantities.

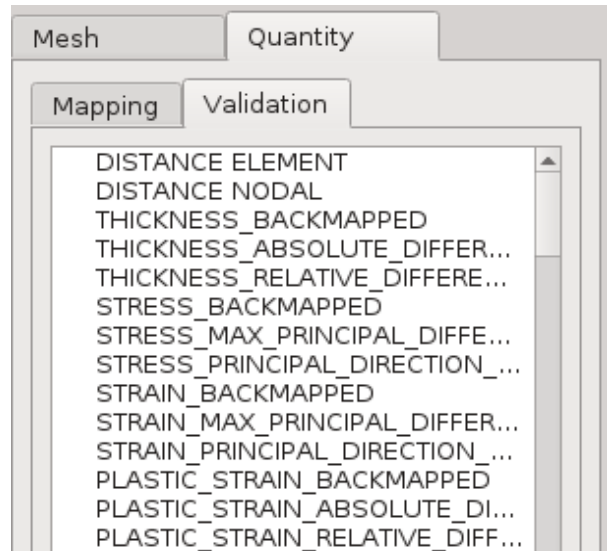


Figure 3.14: Quantity Validation Panel

DISTANCE ELEMENT

Shows the association distance of the current mapping position. The distance of the visible target parts to the visible source parts is calculated.

DISTANCE NODAL

Shows the distance to the next node of the current mapping position. The distance of the visible target parts to the visible source parts is calculated.

Scalar Quantities

During validation of a scalar quantity three validation quantities are created. First, a quantity containing the result of mapping information back from target to source mesh is generated having extension “_BACKMAPPED”. Second, using base and “_BACKMAPPED” quantity, absolute difference values are computed and stored having “_ABSOLUTE_DIFFERENCE” extension. Third, using base and “_ABSOLUTE_DIFFERENCE” quantity absolute relative difference values are computed and stored having “_RELATIVE_DIFFERENCE” extension.

Vector Quantities

During validation of a scalar quantity three validation quantities are created. First, a quantity containing the result of mapping information back from target to source mesh is generated having extension “_BACKMAPPED”. Second, using base and “_BACKMAPPED” quantity, absolute difference values are computed per vector component and stored having “_ABSOLUTE_DIFFERENCE” extension. Third, using base and “_BACKMAPPED” quantity, the angular deviation in degree is computed and stored having “_DIRECTION_DIFFERENCE” extension.

Tensor Quantities

During validation of a scalar quantity three validation quantities are created. First,

a quantity containing the result of mapping information back from target to source mesh is generated having extension “_BACKMAPPED”. Second, using base and “_BACKMAPPED” quantity, absolute difference of tensor principal values are computed and stored having “_MAX_PRINCIPAL_DIFFERENCE” extension. Third, using base and “_BACKMAPPED” quantity, the angular deviation in degree is computed for all principal directions and stored having the extension “_PRINCIPAL_DIRECTION_DIFFERENCE”.

Supplementary to the automatic mode, individual quantity sets can be defined using right click option on validation list and selecting “Validate Quantity” from context menu. In the following “Quantity Validation Dialog” 3.15 available quantities can be assigned to validation list.

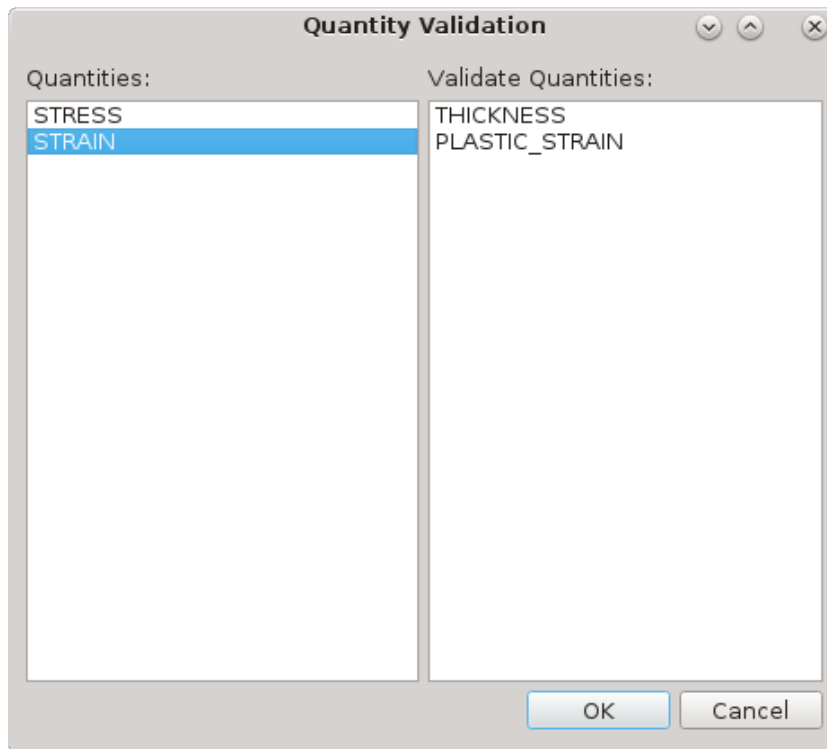


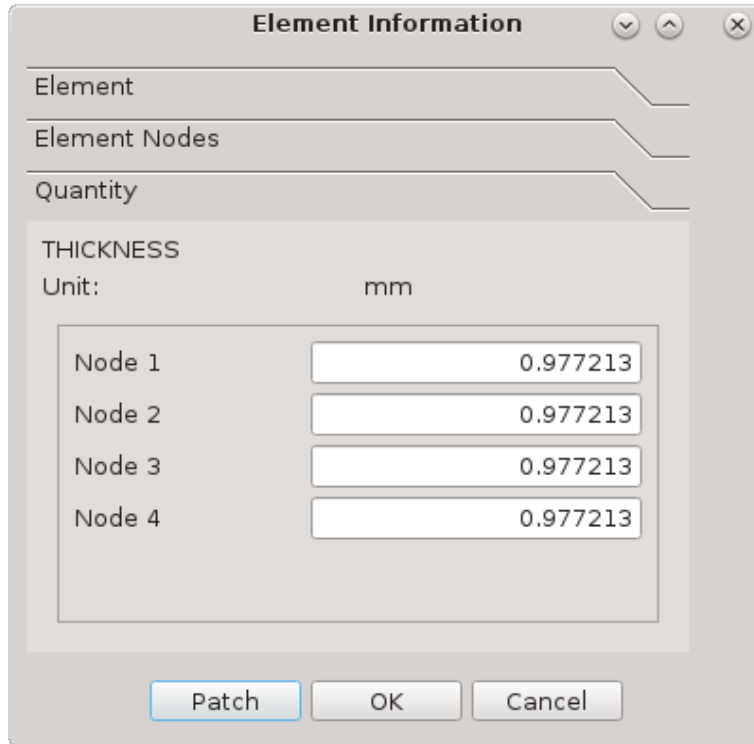
Figure 3.15: Quantity Validation Dialog

3.5 Postprocessing

3.5.1 Element Information

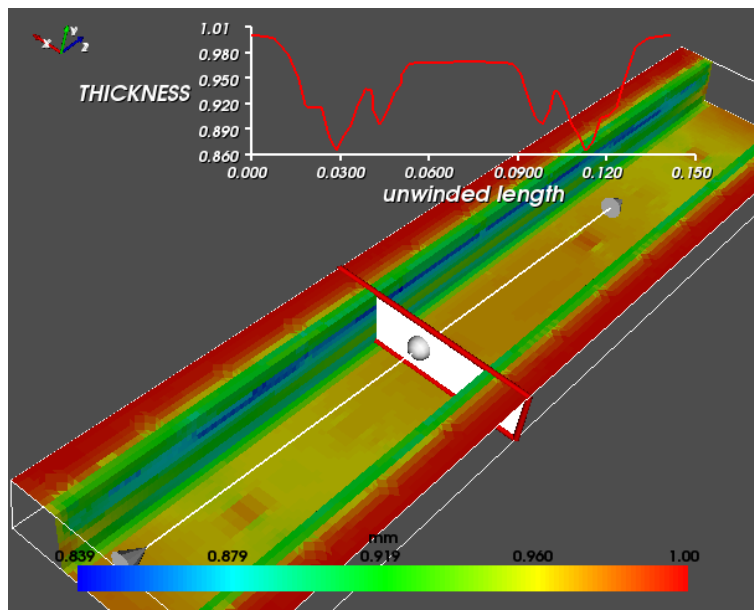
By pressing the “element information button” located at the toolbar and picking an element of a mesh in the viewport it is possible to access geometric and (scalar) quantity information of this element. As graphical assistance the selected element is bordered in the visualisation window. The new information window lists:

- Id of selected element
- Type of selected element
- Ids of nodes forming the element
- Scalar value on element / each node are show and can be edited
- Current unit of active quantity



3.5.2 Slice Model and Plot

Pressing the "slice model plot" of the toolbar an interactive plane widget will be shown in the viewport. By manual interaction with the plane normal arrow, the position and direction of a slice through the model can be defined.



x-axis: unwinded length of the traverse

y-axis: interpolated quantity value

Basic interactions on plane widget

- Recompute values by click on plane surface
- Rotate plane normal by left click on arrow and mouse movement

- Translate normal rotation point by middle click on ball and manual positioning
- Translate plane by either left or middle click on plane
- Shrink or extend bounding box by right click and mouse movement

For every position update of the plane surface the MpCCI Mapper tries to compute a so called length plot corresponding to the quantity distribution. Thereby element edges are intersected with the plane representation and nodal or elemental scalar data is interpolated on the intersection points. Then a traverse is computed that represents the geometry along the defined plane.

Notes and restriction

Multiple models can simultaneously be cut using the slice functionality; multiply plots get an x-axis offset corresponding to the geometric distance of the traverse starting point. Plot axis can get manually scaled at the end of the axis (cursor changes).

3.6 Mapping

A complete mapping cycle between one source and one target model can be done with just a few clicks. The following complete mapping and validation workflow illustrates the typical use of MpCCI Mapper tool. Most of the described steps are optional and may depend on the models and detail of mapping analysis.

Complete Mapping and Validation Workflow

1. Load a source mesh
2. Load a target mesh
3. Switch to mesh panel (optional)
4. Select source and target mesh parts that shall be used for mapping (optional)
5. Align models (optional)
6. Switch to quantity mapping panel (optional)
7. Select quantities which shall be mapped on target (optional)
8. Press mapping button
9. Switch to quantity validation panel (optional)
10. Press validation button for quantity validation (optional)
11. Save target to file

Restrictions:

1. Only quantities are mapped that are available on the source and can be created (if not present) on target.
2. No iterative mapping is supported. If you want to map several sources on one target you need to map all at the same time.
3. Mapping solid to shell models requires only one source and one target mesh.
4. Integral values cannot be conserved.
5. Out of plane integration rules are defined for Gauss up to 16, Lobatto up to 16, Trapezoidal up to 15, Simpson up to 9 integration points.
6. For stresses only nearest data point should be used (c.f. Mapping Parameter Section 3.1.2).

3.7 Batch Mode

MpCCI Mapper allows direct loading, mapping of models and result export via program argument list. The required file readers are automatically detected by file extension.

By default the mapping in batch mode does require same local coordinate system for source and target model. In cases where the automatic alignment tools of MpCCI Mapper (c.f. 3.4.1) produces an acceptable alignment, these automatic positioning can be used in batch as well. If automatic alignment does not work, it is possible to use exported model transformation from manual alignment.

List of MpCCI Mapper argument list options:

-nogui

Enable batch mode of MpCCI Mapper

-csv2dyna

Use Autoform CSV output to LSDyna keyword conversion tool

-csv2vtk

Use CSV to VTK legacy conversion tool

-disablestress

Disable mapping of quantity STRESS

-maximalsearchdistance

Define the maximal search distance parameter (in model length unit) (c.f. 3.5)

-configfile

Option to define program settings based on configuration file (c.f. 3.7.5)

-help

This screen

List of MpCCI Mapper arguments for automatic alignment:

-align

Enable automatic (coarse + fine) model alignment

-coarsealign

Enable coarse model alignment

-finealign

Enable fine model alignment

-inversealign

Enable automatic (coarse + fine) model alignment interchanging source and target model

-inversecoarsealign

Enable coarse model alignment interchanging source and target model

-inversefinealign

Enable fine model alignment interchanging source and target model

-applytransform

Enable model alignment using predefined transformation matrix (c.f. 3.4.1)

3.7.1 Batch on Linux Cluster

The graphical user interface of MpCCI Mapper is based on the QT GUI-Toolkit. Its implementation on Linux does require an active X Window System framework even if the program is running in batch. Normally, Linux cluster compute nodes do not allow the execution of graphical applications using X Window System. Therefore exist virtual display server, e.g. Xvfb (X virtual framebuffer), implementing the X11 display server protocol. Xvfb performs all graphical operations in virtual memory without showing any screen output.

Before execution of MpCCI Mapper in a shell script on a cluster compute node, a virtual display server can be started by adding the lines

```
Xvfb :1 &  
export DISPLAY=:1
```

previous to the `mapper.exe` program call.

3.7.2 Examples

MpCCI Mapper examples using automatic model loading, mapping and result export:

1. `mapper.exe source_file`
Starts GUI and loads `source_file` as source model into MpCCI Mapper .
2. `mapper.exe source_file target_file`
Starts GUI and loads `source_file` as source model and `target_file` as target model into MpCCI Mapper and maps all defined quantities from source to target.
3. `mapper.exe source_file target_file result_file`
Same as 2); the mapped results are then exported into `result_file` in the default export format of `target_file`.

3.7.3 Batch Mode Example

To perform an automatic model import and mapping in batch MpCCI Mapper add the `-nogui` option as first argument:

4. `mapper.exe -nogui source_file target_file result_file`
Same as 3) but the MpCCI Mapper GUI is disabled during the whole process.
5. `mapper.exe -nogui -config config_file source_file target_file result_file`
Same as 4) but the MpCCI Mapper GUI parameter are set using configuration file.

3.7.4 Alignment Examples

To make use of automatic positioning add one of the options `-align`, `-coarsealign`, `-finealign` or their inverse to argument list, e.g

- 2b. `mapper.exe -align source_file target_file`
- 3b. `mapper.exe -finealign source_file target_file result_file`

4b. `mapper.exe -nogui -align source_file target_file result_file`

3.7.5 Batch Configuration File

Beside starting the MpCCI Mapper in batch with additional process control commands, the complete mapping setup can be specified using a text based configuration file.

*optionAlign

Define automatic model alignment method (same as the options in 3.7)

Value	Description
none	Models are aligned (Default)
-align	Enable automatic (coarse + fine) model alignment
-coarsealign	Enable coarse model alignment
-finealign	Enable fine model alignment
-inversealign	Enable automatic (coarse + fine) model alignment interchanging source and target model
-inversecoarsealign	Enable coarse model alignment interchanging source and target model
-inversefinealign	Enable fine model alignment interchanging source and target model

*optionTransformation

Define transformation file for model alignment (same as "-applytransform" in 3.7)

Value	Description
none	default
Global_Path_To_File/transform.trf	

*optionMappingDefault

Define parameter "Default" in Mapping Settings 3.1.2

Value	Description
4	Default
1-8	Specify number of data points to use for interpolation

*optionMappingStressOrHistory

Define parameter "STRESS/HISTORY" in Mapping Settings 3.1.2

Value	Description
4	Default
1-8	Specify number of data points to use for interpolation

*optionMappingOrientations

Define parameter "Orientations" in Mapping Settings 3.1.2

Value	Description
1	Default
1-8	Specify number of data points to use for interpolation

*optionMappingMaximalSearchDistance

Define parameter "Maximal Search Distance" in Mapping Settings 3.1.2

Value	Description
-1	No maximal search distance (Default)
$d > 0$.	Maximal search distance d should be used

***optionMappingMATFEMSettings**

Enable special handling of LSDyna history variables when mapping a MATFEM GenYld

- CrachFEM result file

Value	Description
0	Disabled (Default)
1	Enabled

***optionReaderDynaMAT249**

Active automatic transformation of ply orientation for history variables of LSDyna
MAT_249 4.7.4

Value	Description
0	Disabled (Default)
1	Enabled

***optionWriterDynaOutput**

Define the mode output files for LSDyna are written

Value	Description
0	Single and multiple file are written (Default)
1	Single file only is written
-1	Multiple files only are written

***optionWriterDynaMAT58FiberMode**

Automatic transformation of ply orientation into MAT_58 4.7.4

Value	Description
0	Align element coordinate system using fiber 1 (Default)
1	Align element coordinate system congruent bisectrix

***optionWriterDynaOrientationToHistory**

Define how variable ORIENTATION_TENSOR from molding simulation is handled on
LSDyna file export

Value	Description
0	Disabled (Default), ELEMENT_SOLID_ORTHO is written from ORIENTATION_TENSOR 4.7.5
1	Enabled, generate MATFEM GenYld - CrachFEM history variables

***optionDiableStress**

Special flag to disable quantity STRESS for mapping

Value	Description
0	Map quantity STRESS (Default)
1	Disable quantity STRESS for mapping

***optionMappingQuantity**

Define a list of quantities to map from source to target

Value	Description
-1	Map all quantities (Default)
#N "Quant_1" ...	Map only the <i>N</i> quantities defined by named list

Example of a configuration file to map only "THICKNESS" and "PLASTIC_STRAIN"
using automatic model alignment:

```
#comment line
*optionAlign
-align
*optionTransformation
none
*optionDiableStress
0      # 0 = off  1 = on
```

```
*optionMappingDefault
4      # 1 to 8 points
*optionMappingStressOrHistory
4      # 1 to 8 points
*optionMappingOrientations
1      # 1 to 8 points
*optionMappingMaximalSearchDistance
-1.    # Only relevant if value greater 0.
*optionMappingThicknessInterpolation
1      # 0 = off  1 = on
*optionMappingMATFEMSettings
0      # 0 = off  1 = on
*optionReaderDynaMAT249
0      # 0 = off  1 = on
*optionWriterDynaOutput
1      # [-1 / 0 / 1] (Split File / Split and Combined Files / Combined File)
*optionWriterDynaMAT58FiberMode
1      # 0 = align X  1 = congruent
*optionWriterDynaOrientationToHistory
1      # 0 = off  1 = on
*optionMappingQuantity
2      "THICKNESS" "PLASTIC_STRAIN"
```

4 Solver Formats

4.1 Abaqus

4.1.1 File Format

Abaqus Input Deck (*.inp) is supported for reading and writing.

The model can consist of several *PART definitions but requires unique node id assignments. For example having two *PART definitions "A", nodes numbered from 1 to 100, and "B", nodes numbered from 1 to 80, will lead to an error due to double defined node ids.

Thus, a renumbering of node ids can be performed using Abaqus CAE.

- Import your model file in Abaqus CAE
- Create a new Job from imported model
- Right-Click on the created Job entry and select 'Write Input'

In MpCCI Mapper multiple instances of a *PART is not supported.

An element-based *SURFACE definition is required 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.

4.1.2 Element Types

Shells

MpCCI Mapper supports reading the following shell element types: S4, S4R, S4RS, S4RS, S3, S3R, S3RS

Solids - Surface Definition

MpCCI Mapper supports reading *SURFACE definition of the following 3D solid elements: DC3D4, C3D4, DC3D6, C3D6, DC3D8, C3D8, DC3D10, C3D10

4.1.3 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	x
STRESS	x	x
PLASTIC_STRAIN	x	x
PRESSURE	-	x
ORIENTATION	-	x
PLY_VECTOR	-	x

Quantity	Boundary Condition Type
THICKNESS	*NODAL THICKNESS
	*DISTRIBUTION, LOCATION=ELEMENT, NAME=SHELL_THICKNESS_DIST
STRESS	*INITIAL CONDITIONS, TYPE=STRESS
PLASTIC_STRAIN	*INITIAL CONDITIONS, TYPE=HARDENING
PRESSURE	*DLOAD
ORIENTATION	*ORIENTATION
	*DISTRIBUTION TABLE using COORD3D, *DISTRIBUTION
PLY_VECTOR	*SHELL SECTION, , COMPOSITE
	*DISTRIBUTION TABLE, *DISTRIBUTION and *ORIENTATION

4.1.4 Output Files

To use the files in your input file include them via the *INCLUDE keyword. To register the nodal thickness add the NODAL THICKNESS option to the relevant *SHELL SECTION entries

e.g.:

```
*SHELL SECTION, NODAL THICKNESS, ELSET=..., SECTION INTEGRATION=SIMPSON, MATERIAL=...
```

To register element thickness use the DISTRIBUTION TABLE option, e.g.

```
*SHELL SECTION, SHELL THICKNESS=SHELL_THICKNESS_DIST, ELSET=..., SECTION INTEGRATION=...
```


*.inp	File with model geometry information Include statements for other result files
*.strain	Equivalent plastic strain as *INITIAL CONDITIONS,TYPE=HARDENING
*.stress	Stress tensor as *INITIAL CONDITIONS,TYPE=STRESS
*.thick	Shell THICKNESS as nodal thickness
*.elemthick	Shell THICKNESS as element based thickness
*.pressure	Element surface pressure as *DLOAD
*.orientation	Element local material directions for anisotropic materials
*.sectioncompositedistribution	Composites *DISTRIBUTION TABLE definition to be included on *ASSEMBLY level
*.shellsectioncomposite	Composites *SHELL SECTION definition to be included on *PART level

Table 4.1: Output files generated from MpCCI Mapper

4.2 ANSYS

4.2.1 File Format

The ANSYS Mechanical APDL model must have common database (*.cdb) format to import into MpCCI Mapper . Only the keywords "NBLOCK", "EBLOCK", "CMBLOCK", "ET" and "KEYOP" are currently supported.

4.2.2 Element Types

Shells

Full integrated trians (3 nodes) and quads(4 nodes).

Solids

Reduced integrated tetrahedrons (4 & 10 nodes), hexahedrons (8 & 20 nodes) and prisms (6 nodes) are supported for reading and writing. Fully integrated elements are converted to reduced integrated elements.

4.2.3 Transferable Quantities

Quantities THICKNESS, TEMPERATURE, STRESS, STRAIN, PLASTIC_STRAIN and PRESSURE are supported for mapping and writing. Boudary condition type INISTATE use defaut coordinate system identifier 0.

Quantity	Boundary Condition Type
THICKNESS	Tabular \$NodeId \$Value
TEMPERATURE	BF, ,TEMP
STRESS	INISTATE, SET, DTYP, S
STRAIN	INISTATE, SET, DTYP, EPEL
PLASTIC_STRAIN	INISTATE, SET, DTYP, EPPL
PRESSURE	SFE, , ,PRES

4.3 AutoForm R4 converter

MpCCI Mapper command to convert exported CSV format reads as follows

```
> mapper.exe -csv2dyna FILE_1 FILE_2 FILE_3 FILE_4
```

where each file in argument list needs to contain parameters as specified in table 4.2.

File	Description
FILE_1	*_NodeData_Thickness containing nodal coordinates and ids
FILE_2	*_ElementData_Thickness containing element topology and thickness
FILE_3	*_ElementData_Plastic_Strain containing element topology plastic strain
FILE_4	Result_File_Name

Table 4.2: AutoForm converter parameter list

Required CSV format per output file:

1. *_NodeData_Thickness:
Node Idx,XCoord,YCoord,ZCoord,,Info
2. *_ElementData_Thickness:
Element Idx,Node Idx 1,Node Idx 2,Node Idx 3,Thickness,,Info
3. *_ElementData_Plastic_Strain (5 points over thickness):
Element Idx,Node Idx 1,Node Idx 2,Node Idx 3,Plastic Strain Layer -2,Plastic Strain Layer -1,Plastic Strain,Plastic Strain Layer 1,Plastic Strain Layer 2,,Info

4.4 Cadmould

4.4.1 File Format

Cadmould geometry file *.cfe is supported. The file contains the element definitions, the node coordinates, the number of nodes and elements and an identifier if it is a surface mesh or a midplane mesh. MpCCI Mapper can import both midplane and surface meshes. Surface meshes consist of pairs of triangles which are internally converted into a layered prism mesh. Midplane meshes consist of triangular shell elements. As result format binary *.car files containing ORIENTATION_TENSOR information can be read.

4.4.2 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	-
ORIENTATION_TENSOR	x	-

4.5 Indeed

4.5.1 File Format

For code Indeed reading of ASCII result file (*.asc) and geometry file (*.inc) is supported.

4.5.2 Element Types

The 6- and 8-node Crisfield-Solid-Shell elements are supported for reading.

4.5.3 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	-
STRESS	x	-
PLASTIC_STRAIN	x	-

4.6 Forge / SIMUFACT

4.6.1 File Format

The Universal File format (*.unv), which can be exported via the IDEAS export interface in GLPre, is supported by MpCCI Mapper .

4.6.2 Element Types

Shells

Full integrated trians (3 & 6 nodes) and quads(4 nodes) are supported for reading.

Solids

Tetrahedrons (4 & 10 nodes), hexahedrons (8 & 20 nodes) and prisms (6 nodes) are supported for reading.

4.6.3 Transferable Quantities

In general nodal based (card format 55), element based (card format 56) and general card format 2414 quantities can be read from Universal File format. Variables TEMPERATURE and STRESSTENSOR are automatically detected and assigned as MpCCI Mapper quantities TEMPERATURE and STRESS.

4.7 LSDyna

4.7.1 File Format

LSDyna keyword format (*.k,*.key,*.dyn,*.dynain) is supported for reading and writing.

4.7.2 Element Types

*ELEMENT_SHELL Card

Full integrated trians (3 nodes) and quads(4 nodes). Arbitrary out of plane integration types up to 16 integration points out of plane.

*ELEMENT_SOLID Card

Tetrahedrons (4 & 10 nodes), hexahedrons (8 nodes) and prisms (6 nodes) are supported for reading and writing. Element formulations -2, -1, 1, 2, 10, 13, 15, 16 and 115 form solid section can be used for quantity generation.

4.7.3 Transferable Quantities

Quantity	Read	Write	Boundary Condition or Element Card
THICKNESS	x	x	*ELEMENT_SHELL_THICKNESS
STRESS	x	x	*INITIAL_STRESS_SHELL
	x	x	*INITIAL_STRESS_SOLID
STRAIN	x	x	*INITIAL_STRAIN_SHELL
	x	x	*INITIAL_STRAIN_SOLID
PLASTIC_STRAIN	x	x	*INITIAL_STRESS_SHELL
	x	x	*INITIAL_STRESS_SOLID
HISTORY Variables	x	x	*INITIAL_STRESS_SHELL
	x	x	*INITIAL_STRESS_SOLID
ORIENTATION	-	x	*INITIAL_STRESS_SHELL
	x	x	*ELEMENT_SOLID_ORTHO
ELEMENT_SHELL_BETA	-	x	*INITIAL_STRESS_SOLID
	x	x	*ELEMENT_SHELL_THICKNESS

Table 4.3: MpCCI Mapper quantities in LSDyna solver format

4.7.4 Composite Material Model Features

For LSDyna Version R 10 MpCCI Mapper supports direct mapping of locally orthotropic material axes from *MAT_REINFORCED_THERMOPLASTIC (*MAT_249) to MAT_LAMINATED_COMPOSITE_FABRIC (*MAT_58). Therefor direction of fibers, stored in certain HISTORY Variables, are gathered in separated vector quantites:

Quantity Name	HISTORY Variable	Comment
PLY_VECTOR_1	POSTV dependent	Direction 1st fiber (global coordinates)
PLY_VECTOR_2	POSTV dependent	Direction 2nd fiber (global coordinates)

Table 4.4: MpCCI Mapper quantities for composites.

4.7.5 Injection Molding Features

The fiber ORIENTATION_TENSOR result shows the probability of fiber alignment in the specified principal direction at the end of a injection molding process. Depending on the degree of orientation MpCCI Mapper can classify and export 15 different placeholder material cards as well as section solid assignments for solid elements. For each solid element local material axes are defined using *ELEMENT_SOLID_ORTHO card. For material #1 a *MAT_ELASTIC template and for material 2-15 a *MAT_ANISOTROPIC_ELASTIC template is written to output files.

Id	Material Title	SECTION_SOLID_TITLE	Comment
1	MATERIAL_ISOTROPIC	PART_ISOTROPIC	a11 = a22 = a33 = 0.33
2	MATERIAL_0_4_0_3_0_3	PART_0_4_0_3_0_3	a11 = 0.4, a22 = 0.3, a33 = 0.3
3	MATERIAL_0_4_0_4_0_2	PART_0_4_0_4_0_2	a11 = 0.4, a22 = 0.4, a33 = 0.3
4	MATERIAL_0_5_0_3_0_2	PART_0_5_0_3_0_2	a11 = 0.5, a22 = 0.3, a33 = 0.2
5	MATERIAL_0_5_0_4_0_1	PART_0_5_0_4_0_1	a11 = 0.5, a22 = 0.4, a33 = 0.1
6	MATERIAL_0_5_0_5_0_0	PART_0_5_0_5_0_0	a11 = 0.5, a22 = 0.5, a33 = 0.0
7	MATERIAL_0_6_0_2_0_2	PART_0_6_0_2_0_2	a11 = 0.6, a22 = 0.2, a33 = 0.2
8	MATERIAL_0_6_0_3_0_1	PART_0_6_0_3_0_1	a11 = 0.6, a22 = 0.3, a33 = 0.1
9	MATERIAL_0_6_0_4_0_0	PART_0_6_0_4_0_0	a11 = 0.6, a22 = 0.4, a33 = 0.0
10	MATERIAL_0_7_0_2_0_1	PART_0_7_0_2_0_1	a11 = 0.7, a22 = 0.2, a33 = 0.1
11	MATERIAL_0_7_0_3_0_0	PART_0_7_0_3_0_0	a11 = 0.7, a22 = 0.3, a33 = 0.0
12	MATERIAL_0_8_0_1_0_1	PART_0_8_0_1_0_1	a11 = 0.8, a22 = 0.1, a33 = 0.1
13	MATERIAL_0_8_0_2_0_0	PART_0_8_0_2_0_0	a11 = 0.8, a22 = 0.2, a33 = 0.0
14	MATERIAL_0_9_0_1_0_0	PART_0_9_0_1_0_0	a11 = 0.9, a22 = 0.1, a33 = 0.0
15	MATERIAL_1_0_0_0_0_0	PART_1_0_0_0_0_0	a11 = 1.0, a22 = 0.0, a33 = 0.0

Table 4.5: *ELEMENT_SOLID_ORTHO output generated by MpCCI Mapper from a mapped ORIENTATION_TENSOR.

4.7.6 Output Files

*.key	File with model *NODE definition Include statements for *.key.thick and *.key.stressstrain
*.key.thick	Model *ELEMENT definition with THICKNESS
*.key.stressstrain	STRESS, STRAIN, PLASTIC_STRAIN and HISTORY Variables
*_complete.key	Combinend *.key, *.key.thick and *.key.stressstrain output
*_with_displacements.key	Optional file containing morphed coordinates when mapping solid to shell

Table 4.6: Output files generated from MpCCI Mapper

Historyvariables

MpCCI Mapper allows a redefinition of the output sequence of history variables via ASCII-based configuration file over the specific environment variable or via the file writer settings dialog 3.1.2. Using the environment variable option, a user has to specify the variable

LSDYNA_HISTORY_CONFIGURATION_FILE=Global_Path_To_File/File

which defines the global path to the configuration file which shall be used for history variable exchange. Starting with keyword “MapperHistoryVariableConfiguration” the configuration is given in tabular format

```
MapperHistoryVariableConfiguration
OriginalHistoryId NewHistoryId ScalingFactor
```

In some cases a mapping of history variables between different material models require an additional scaling of a variable. To scale a variable on result export an auxiliary scaling factor can be specified in column number three. As default a scaling factor of 1.0 is applied if the column entry is not specified.

Example:

```
MapperHistoryVariableConfiguration
1 1
2 4 1.5
3 3
4 2
5 5 3.0
6 7
7 6
8 8
```

Figure 4.1 shows the history variable handling when writing a LSDyna model. Here Variable 2 and 4 as well as 6 and 7 are swapped in output order. Variable 2 is additionally scaled by a factor of 1.5, variable 5 will be scaled by a factor of 3.0.

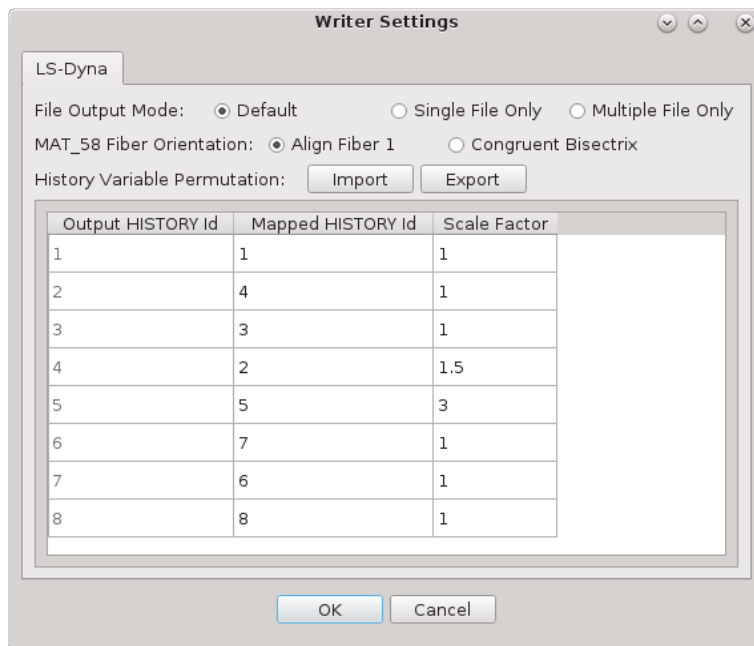


Figure 4.1: History variable permutation imported in writer settings dialog.

The same output can be generated using short notation

```
MapperHistoryVariableConfiguration
2 4 1.5
4 2
```

5 5 3.0
6 7
7 6

If the new history id is set to "0", the variable is set to zero for all integration points and thus treated as empty.

4.8 Marc / COPRA FEA

4.8.1 File Format

MpCCI Mapper supports reading of ASCII based MSC Marc result format (*.t19). By default MSC Marc does export binary .t16 files only which are not supported. To request an ASCII .t19 select the relevant Job in MSC Mentat Preprocessor, and go 'Job Properties' → 'Job Results' → 'Post File' and activate 'Binary & ASCII' option. Then, both ASCII and Binary file will be written during simulation.

4.8.2 Element Types

Mapper Element Type	MSC Marc Element Types
SHELL QUAD 4	3,11,18,22,39,75,85,114,115,139,140,198
SHELL TRIAN 3	6,37,50,138,158,196,210
SHELL QUAD 8	30,32,41,72,86
SHELL TRIAN 6	49,124,125
SOLID TETRA 4	134,135,157
SOLID TETRA 10	127,130,133,184
SOLID HEXA 8	7,43,109,113,117,120,123
SOLID HEXA 20	21,44
SOLID WEDGE 6	136,137,204

Table 4.7: Supported MSC Marc element types

4.8.3 Transferable Quantities

MpCCI Mapper supports reading of nodal displacements and computes deformed geometry for initial definition of nodal coordinates.

Quantity	Read	Write
TEMPERATURE	x	-
STRESS	x	-
STRAIN	x	-
PLASTIC_STRAIN	x	-

4.9 Moldflow

4.9.1 File Format

Moldflow geometry file exported to PATRAN neutral (*.pat) is supported. The file contains the element definitions and the node coordinates. ORIENTATION_TENSOR results as generic .xml or PATRAN neutral (*.nod) are supported for reading.

4.9.2 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	-
ORIENTATION_TENSOR	x	-

4.10 Nastran

4.10.1 File Format

MpCCI Mapper supports reading of ASCII Nastran Bulk data format (*.nas,*.bdf). Nodal coordinates can be either in long (GRID*) or short (GRID) format. Nastran so called free format is not supported.

4.10.2 Element Types

Triangles CTRIA3 and quads CQUAD4 are supported for reading and writing.

4.10.3 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	x
STRESS	-	x
PLASTIC_STRAIN	-	x
PRESSURE	x	x

Quantity	Boundary Condition Type
THICKNESS	element description card format
STRESS	ISTRSSH for SOL 700 only
PLASTIC_STRAIN	ISTRSSH for SOL 700 only
PRESSURE	PLOAD2

4.11 PAM

4.11.1 File Format

MpCCI Mapper supports the following native PAM solver ASCII formats:

PAMStamp mapping files (*M00-*M99)

PAMCrash input deck (*.pc,*.ps)

4.11.2 Element Types

Full integrated triangles (3 nodes) and quads(4 nodes) with arbitrary out of plane integration number can be read. Only active parts are written into *M00-*M99 format for both native solver input types.

4.11.3 Transferable Quantities

Quantity	Read	Write	Boundary Condition
THICKNESS	x	x	THIC
STRESS	x	x	STRS
STRAIN	x	x	STRN
PLASTIC_STRAIN	x	x	PLAS
PRESSURE	-	x	PREFA
HV__	x	x	HV__

4.12 Radioss

4.12.1 File Format

Radioss Result (*Yxxx) Reading and writing is supported

Radioss Result (*.sta) Reading and writing is supported

Radioss Input (*D00, .rad) Reading is supported

4.12.2 Element Types

Fully integrated and reduced integrated triangles and quads are supported.

4.12.3 Transferable Quantities

Quantity	Read	Write
THICKNESS	x	x
STRESS	x	x
PLASTIC_STRAIN	x	x

4.13 Sysweld

4.13.1 File Format

<code>*.asc</code>	Model information file
<code>*_trans.asc</code>	Mechanical or thermal result file Stresses, strain, plastic strain and young modul is extracted. The effectiv plastic strain is extracted from plastic strain.
<code>*_phase.asc</code>	Defining names for the phases.

The syntax is that of the sysweld files and looks like this

<code>SYSWELD_PHASE_DEFINITION</code>	First line of file is the magic word.
<code>BEGIN_PHASEDEF</code>	Starting to define the mnemonics
<code>1 "MARTENSIT"</code>	Name of first phase
<code>2 "AUSTENIT"</code>	Name of second phase
<code>5 "SOMEOTHERUSERDEFINEDPHASE"</code>	Name of phase five
<code>END_PHASEDEF</code>	End of definition statments

Output The files expt the `*_phase.asc` from above are written or are patched where values have changed.

!! Attention, currently the writer is designed to write the mapped values of a metal forming result. To keep equilibrium the elastic strains are calculated from the residual stresses. As only plastic strains can be written and as the total strain needs to be 0 (displacements are assumed to be 0) the plastic strain is set to the calculated negative strain values. The young modulus is used to calculate the strains.

5 Other Formats

5.1 Argus / Aramis

Argus and Aramis offer a configurable output format which can be imported in MpCCI Mapper (*.txt). Thickness information and strain can be imported.

5.1.1 File Format

To define an initial thickness (here 1.00 mm) use following string in file header comment block:

```
# Ausgangsblechdicke: 1.00mm
```

1	2	3	4	5	6	7	8	9	10
s	u	v	x_coord	y_coord	z_coord	phi_1	phi_2	phi_3	%thickness reduction

Table 5.1: Order of Variables required for MpCCI Mapper

5.2 Atos

5.2.1 File Format

The 'Gesellschaft fr Optische Messtechnik' offers an export plugin that allows geometry and thickness information export in polygon file format (*.ply) which is supported for reading in MpCCI Mapper . This plugin can be received directly from GOM <http://www.gom.com>.

5.3 AutoGrid

5.3.1 File Format

MpCCI Mapper supports AutoGrid *.dat file import for software version 4.1.x.x and 4.2.x.x. Thickness and strain can be imported.

5.4 STL

Eighter ASCII or binary STL-files can be imported in MpCCI Mapper using the common *.stl file extension.

5.5 VTK

ASCII based VTK legacy files (*.vtk) can be imported in MpCCI Mapper .

5.5.1 Element Types

VTK	MpCCI Mapper	Comment
VTK_TRIANGLE	TRIAN	3-node shell element
VTK_QUAD	QUAD	4-node shell element
VTK_QUADRATIC_TRIANGLE	TRIAN6	6-node shell element
VTK_QUADRATIC_QUAD	QUAD8	8-node shell element
VTK_TETRA	TETRA	4-node volume element
VTK_HEXAHEDRON	HEXA	8-node volume element
VTK_WEDGE	WEDGE	6-node volume element
VTK_QUADRATIC_TETRA	TETRA10	10-node volume element
VTK_QUADRATIC_HEXAHEDRON	HEXA20	20-node volume element

Table 5.2: Translation of VTK elements in MpCCI Mapper

6 Thirdparty License Information

6.1 LibQxt

Copyright (c) 2006 - 2011, the LibQxt project. See the Qxt AUTHORS file for a list of authors and copyright holders. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the LibQxt project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ;COPYRIGHT HOLDER; BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<http://libqxt.org> foundation@libqxt.org

6.2 Libtiff

Copyright (c) 1988-1997 Sam Leffler Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTIC-

ULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

6.3 Libwebp

Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.4 LibXBC

Copyright (c) 2000 Keith Packard

Copyright (c) 2006 Jamey Sharp

Copyright (c) 2007-2008 Vincent Torri vtorri@univ-evry.fr

Copyright (c) 2007 Bart Massey

Copyright (c) 2008-2009 Julien Danjou julien@danjou.info

Copyright (c) 2008 Arnaud Fontaine arnaud@debian.org

Copyright (c) 2008 Bart Massey bart@cs.pdx.edu

Copyright (c) 2008 Ian Osgood iano@quirkster.com

Copyright (c) 2008 Jamey Sharp jamey@minilop.net

Copyright (c) 2008 Josh Triplett josh@freedesktop.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors or their institutions shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the authors.

6.5 PCRE Library

Copyright (c) 1997-2016 University of Cambridge Copyright (c) 2009-2016 Zoltan Herczeg Copyright (c) 2007-2012 Google Inc. Copyright (c) 2013-2013 Tiler Corporation (jiwang@tilera.com)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University of Cambridge nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.6 VTK

Copyright (c) 1993-2008 Ken Martin, Will Schroeder, Bill Lorensen
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.