1 Release Notes

1.1 Version 4.6.0

List of changes and improvements since MpCCI Mapper version 4.5.3

1. New mapping algorithm for quantity ORIENTATION_TENSOR, based on Log-Euclidean framework, see 3.6.2 and 3.18 for detailed description
2. Quantity validation has been extended by a statistic report measuring minimal and maximal differences in data together with an arithmetic mean and standard deviation information 3.4.3
3. Quantity plot feature added to display the relative distribution of a quantity 3.5.3
4. Change display of quantity units in toolbar quantity dialog and quantity range clip dialog 3.2
5. Transformation matrix storage option for LSDyna added; create INCLUDE_TRANSFORM card for source or target model 3.4.1
6. Disable or enable mapping quantities in quantity context menu 3.4.2
7. Pyramid element with five nodes has been added to MpCCI Mapper

Table 1.1: GUI related changes and new features

4. Solvers MSC.Simufact Welding, Forming and Additive 4.6
   4.1. Universal file format redesigned as standalone interface
   4.2. Multiple bodies in single model now supported
   4.3. Support for layered solid elements
   4.4. Card 2414 format extended for data on integration points
   4.5. Automatic detection of STRESS, PLASTIC_STRAIN, TEMPERATURE and DISPLACEMENTS
5. Solver LS-Dyna 4.8
   5.1. Export of DISPLACEMENTS as deformed geometry and export nodal TEMPERATURE
   5.2. Enhanced support for MATFEM model material orientations
   5.3. Redefinition of element formulation (Elform) and through thickness integration scheme for target models 4.3
6. Solver Moldflow 4.10
   6.1. Geometry model file *.udm supported
   6.2. Import multiple *.xml on file open process
   6.3. Transient *.xml files supported; last defined state is used for import
   6.4. Optimized IO performance on XML data
7. Solver ANSYS Mechanical APDL 4.2
   7.1. List of supported solid elements has been extended
   7.2. INISTATE result format redesigned
   7.3. Export of DISPLACEMENTS as nblock with deformed nodal coordinates
8. Vendor-neutral standard for CAE data storage format VMAP added https://www.vmap.eu.com

Table 1.2: Solver formats changes and extension
1.2 Version 4.5.3

List of changes and improvements since MpCCI Mapper version 4.5.2

1. Added selection dialog if file extension matches more than one reader interface.
2. Model symmetry definition on source models to build symmetric part before mapping.
3. Vector data display has been revised. Data is automatically scaled to average element size. Only quantity DISPLACEMENTS is shows with actual length.
4. Element slice tool fixed for multiple part models.
5. QT library has been update to version 5.9.7.

Table 1.3: GUI related changes and new features

6. Radioss state file has been set as default export format.
7. LS-Dyna variable TEMPERATURE support on export. INITIAL\_STRAIN with variable number of points through shell thickness support.
8. Universal file format (Forge, Simufact) unit system detection. Fixed card 2414 format import on multiple data definition.

Table 1.4: Solver formats changes and extension

1.3 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.
2. Validation concept has been redesigned and is now available for all quantities.
4. Enhanced data visualization for tensor and vector quantities.

Table 1.5: GUI related changes and new features

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

Table 1.6: 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:

```sh
$ hostname
myHostName
```
2.2.3 Configure a License Manager as Windows service

Execute the `lmtools.exe` application from the license manager installation directory:

\[ < \text{LICENSE\_TOOL\_INSTALLATION} > /bin/lmtools.exe \]

- 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 mpccl.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.

  ![Image of LMTOOLS by Acessso Software Inc.](image)

• 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' where basic operations of the MpCCI Mapper can be accessed. Right below the toolbar, the main viewport handles the display of model geometry and quantities in a OpenGL rendering area. Alongside of the viewport, the data panel 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 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. 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 offers several options
for interaction with local file system. A detailed description on each option is given in table 3.1.

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

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. It is designed as a drop down menu containing certain alignment, mapping algorithm and file writer configuration options.

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 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

![Mapping Settings dialog](image)

(1) Default k-nearest-node mapping algorithm parameter \( k \)
(2) Parameter \( k \) for quantity STRESS or LS-Dyna HISTORY variables
(3) Parameter \( k \) for orientations (material direction, fiber direction, ORIENTATION_TENSOR)
(4) Maximal search distance for nearest-node search, negative value means unlimited
(5) Maximal search distance unit
(6) Enable or disable through thickness direction interpolation for shell elements
(7) Solid to shell mapping, translation factor of solid outer surface
(8) Enable or disable distance weighting for Log-Euclidean metric mapping algorithm (ORIENTATION_TENSOR only)

MpCCI Mapper uses a k-nearest-node mapping algorithm (default \( k = 4 \)) that combines robustness with data smoothing. At times were 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 3.1.2, 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 os 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 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 3.1.2.

**Writer Settings**

![Writer Settings](image)

Figure 3.6: Writer settings dialog

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 4.8.6 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.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Load a new source model from file</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Load a new target model from file</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Save active (visible) target models</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Reset Camera to show all active models</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Adjust the quantity color range by the user or reset to automatic range</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Clip a scalar value range&lt;br&gt;Values above and below new limits are set to lower resp. upper limit</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Show info about selected element 3.5.1</td>
</tr>
<tr>
<td><img src="image" alt="icon" /></td>
<td>Slice through model and plot scalars along traverse 3.5.2</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spinning of models</td>
<td>Press and hold left mouse button on a model, then move mouse to rotate</td>
</tr>
<tr>
<td>Translating models</td>
<td>Press and hold middle mouse button on a model, then move mouse to desired location</td>
</tr>
<tr>
<td>Zooming models</td>
<td>Press and hold right mouse button on a model, then move mouse upwards to zoom in or downwards to zoom out</td>
</tr>
<tr>
<td>Rotate view</td>
<td>Press and hold left mouse button on background or hold ‘Shift’ button</td>
</tr>
<tr>
<td>Translating view</td>
<td>Press and hold middle mouse button on background or hold ‘Shift’ button</td>
</tr>
<tr>
<td>Zooming view</td>
<td>Press and hold right mouse button on background or hold ‘Shift’ button</td>
</tr>
</tbody>
</table>

Table 3.2: View and model interaction

3.3.2 Viewport Hotkeys

<table>
<thead>
<tr>
<th>Hotkey</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘w’</td>
<td>Wireframe mode: active parts are presented as wireframes</td>
</tr>
<tr>
<td>‘s’</td>
<td>Surface mode: active parts are presented as surfaces</td>
</tr>
<tr>
<td>‘b’</td>
<td>Surface with mesh wireframe (‘w’+’s’)</td>
</tr>
<tr>
<td>‘p’</td>
<td>Points mode: active parts are presented as point clouds</td>
</tr>
<tr>
<td>‘o’</td>
<td>Opaque mode: toggle opacity/transparency of models</td>
</tr>
<tr>
<td>‘Shift’</td>
<td>Disable model interaction on rotation, translation and zoom</td>
</tr>
</tbody>
</table>

Table 3.3: List of hotkeys for viewport interaction

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 into 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

![Image of 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](image)

**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](image)

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).

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember current position for mapping and validation</td>
</tr>
<tr>
<td>Set model positions to last remembered position.</td>
</tr>
<tr>
<td>Rough Align:</td>
</tr>
<tr>
<td>Used to bring equal meshes close to each other.</td>
</tr>
<tr>
<td>Fine Align:</td>
</tr>
<tr>
<td>Align meshes for mapping. Can be used several times to improve the result.</td>
</tr>
<tr>
<td>Symmetric Align:</td>
</tr>
<tr>
<td>Used if one of the models uses axes symmetry</td>
</tr>
<tr>
<td>Mirror Geometry:</td>
</tr>
<tr>
<td>Mirrors all active models</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>Save current positions to a file:</td>
</tr>
<tr>
<td>1. It is possible to store the transformations in Mapper format (*.trf)</td>
</tr>
<tr>
<td>2. Export target mesh nodes in current position in LSDyna keyword format</td>
</tr>
<tr>
<td>3. Export source mesh nodes in current position in LSDyna keyword format</td>
</tr>
<tr>
<td>4. Export target mesh position using INCLUDE_TRANSFORM LSDyna card</td>
</tr>
<tr>
<td>5. Export source mesh position using INCLUDE_TRANSFORM LSDyna card</td>
</tr>
<tr>
<td>Load transformation from a file:</td>
</tr>
<tr>
<td>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 &quot;*transformation&quot;. Declaration of the transformation begins with a new line and is read linewise; entries need to be separated by blanks.</td>
</tr>
<tr>
<td>Note: All active targets will be transformed.</td>
</tr>
</tbody>
</table>

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 $A = (a_{ij})_{i,j=1,...,3}$ is the rotation matrix of a model in the 3-dimensional space and $T = (t_i)_{i=1,...,3}$ 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
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-9.999999e-01</td>
<td>3.289400e-05</td>
<td>5.307189e-04</td>
<td>1.028435e+00</td>
<td></td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0.000000e+00</td>
<td>0.000000e+00</td>
<td>1.000000e+00</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Left Click</th>
<th>Select/deselect quantity for visualization of values as contour plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double (Left or Mid) Click</td>
<td>Activate/deactivate quantity for mapping (black/grey letters)</td>
</tr>
<tr>
<td>Right Click</td>
<td>Open quantity dialog box 3.4.2</td>
</tr>
<tr>
<td>Deselect</td>
<td>Show parts</td>
</tr>
</tbody>
</table>

Table 3.5: Mapping quantity interaction

Remark:

Averaged values of all element integration points are used for element based data visualisation. 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).

---

Activate

Activate quantity for mapping

Deactivate

Deactivate quantity for mapping

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
3.4.3 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.

**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” available quantities can be assigned to validation list.

After the (mapping) validation has been performed, a statistical analysis of the obtained validation quantities can be displayed. Here, for each validation quantity the absolute minimum and maximum as well as the arithmetic mean together with the standard deviation is computed. At the end, those data is visualized in a tabular report dialog. Optionally the table result can be exported as comma separated value (*.csv) file.
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 in 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.

![Slice Model and Plot](image)

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 simultaniously 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).

3.5.3 Quantity Plot

To study the relative distribution of a quantity, it’s distribution on a percentage basis can be visualized as 2D-plot. First, select desired quantity in "Mapping" or "Validation" tab
of the quantity panel. On right-click the quantity name the context menu opens then select "Plot Quantity" option in list.

![Curve Plot Dialog](image)

Figure 3.17: Distribution of a certain quantity

The value range interval will be divided in ten equally spaced sub-intervals and the relative frequency of data points in the sub-interval range is plotted for each model separately.

### 3.6 Mapping - Theory of Algorithms in MpCCI Mapper

The discrete mapping problem for a set of \( N \) known data points, described as a list of tuples \( \{(x_1, u_1), (x_2, u_2), \ldots, (x_N, u_N)\} \), where \( x_i \) denote points in the 3-dimensional space and \( u_i \) the corresponding values of an unknown function \( u(x) : x \rightarrow \mathbb{R} \), \( x \in \mathbb{D} \subset \mathbb{R}^3 \) Goal of the interpolation problem is to find a function \( \tilde{u} \) to be "smooth" and to be exact \( \tilde{u}(x_i) = u_i \).

#### 3.6.1 Euclidean inverse distance weighted k-nearest

A general form of finding an interpolating function \( \tilde{u} \) at a given point \( x \) based on samples \( \tilde{u}(x_i) = u_i \) for \( i = 1, \ldots, N \) is the Euclidean Inverse Distance Weighting method

\[
\tilde{u}(x) = \begin{cases} 
\sum_{i=1}^{N} \frac{w_i(x)u_i}{\sum_{i=1}^{N} w_i(x)}, & \text{if } ||x - x_i|| > 0 \text{ for all } i \\
u_i, & \text{if } ||x - x_i|| = 0 \text{ for some } i
\end{cases}
\]

where

\[
w_i(x) = \frac{1}{||x - x_i||^p}
\]

and \( p \) is a positive real number, called the power parameter. As the interpolation function depends \( \tilde{u} \) on all data tuples, its evaluation gets more and more time consuming with increasing number of \( N \). By construction of Inverse Distance Weighting, the influence - meaning the weight - of one data pair \( (x_i, u_i) \) decreases exponentially with the Euclidean distance to \( x_i \).

Utilizing the exponential decrease of the weighting function, the above interpolation scheme can be modified by the k-nearest approach

\[
w_i(x) = \begin{cases} 
\frac{1}{||x - x_i||^p}, & \text{for those indexes } i \text{ having the k-th minimal distance to } x \\
0, & \text{else}
\end{cases}
\]
3.6.2 Log-Euclidean weighted k-nearest

The exponential of a matrix $A$ is defined by

$$exp(A) = \sum_{i=1}^{\infty} \frac{A^n}{n!}.$$  

To a given matrix $A$, another matrix $B$ is said to be the matrix logarithm of $A$ if $exp(B) = A$ is satisfied.

Log-Euclidean metric for symmetric positive definite (spd) matrices

$$dist(S_1, S_2) = \|log(S_1) - log(S_2)\|$$

builds an isomorphism (the algebraic structure of a space is conserved) and is an isometry (distances are conserved) between the space of spd matrices and the Euclidean space.

Having a set of $N$ tensors $S_1, ..., S_N$ with arbitrary positive weights $w_1, ..., w_N$ we can formulate the Log-Euclidean Fréchet mean of $N$ tensors by

$$E(S_1, ..., S_N) = exp(\sum_{i=1}^{N} w_i log(S_i))$$

A comparison of the Euclidean and the Log-Euclidean interpolation result of tensors is illustrated in figure 3.18.

Figure 3.18: Bilinear interpolation of 4 tensors at the corners of a grid. Left figure: Euclidean interpolation Right figure: Log-Euclidean interpolation.

---

1 Images from Fast and Simple Calculus on Tensors in the Log-Euclidean Framework, Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache
3.7 Mapping with Mapper

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, Trapezodial 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]).
3.8 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. 3.1.2)

- **-configfile**
  Option to define program settings based on configuration file (c.f. 3.8.5)

- **-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)
3.8.1 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.

3.8.2 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`.

3.8.3 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.

3.8.4 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`

### 3.8.5 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 3.8)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Models are aligned (Default)</td>
</tr>
<tr>
<td>-align</td>
<td>Enable automatic (coarse + fine) model alignment</td>
</tr>
<tr>
<td>-coarsealign</td>
<td>Enable coarse model alignment</td>
</tr>
<tr>
<td>-finealign</td>
<td>Enable fine model alignment</td>
</tr>
<tr>
<td>-inversealign</td>
<td>Enable automatic (coarse + fine) model alignment interchanging source and target model</td>
</tr>
<tr>
<td>-inversecoarsealign</td>
<td>Enable coarse model alignment interchanging source and target model</td>
</tr>
<tr>
<td>-inversefinealign</td>
<td>Enable fine model alignment interchanging source and target model</td>
</tr>
</tbody>
</table>

*optionTransformation*

Define transformation file for model alignment (same as ”-applytransform“ in 3.8)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>default</td>
</tr>
<tr>
<td>Global_Path_To_File/transform.trf</td>
<td></td>
</tr>
</tbody>
</table>

*optionMappingDefault*

Define parameter ”Default“ in Mapping Settings [3.1.2]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Default</td>
</tr>
<tr>
<td>1-8</td>
<td>Specify number of data points to use for interpolation</td>
</tr>
</tbody>
</table>

*optionMappingStressOrHistory*

Define parameter ”STRESS/HISTORY“ in Mapping Settings [3.1.2]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Default</td>
</tr>
<tr>
<td>1-8</td>
<td>Specify number of data points to use for interpolation</td>
</tr>
</tbody>
</table>

*optionMappingOrientations*

Define parameter ”Orientations“ in Mapping Settings [3.1.2]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td>1-8</td>
<td>Specify number of data points to use for interpolation</td>
</tr>
</tbody>
</table>

*optionMappingMaximalSearchDistance*

Define parameter ”Maximal Search Distance“ in Mapping Settings [3.1.2]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>No maximal search distance (Default)</td>
</tr>
<tr>
<td>$d &gt; 0.$</td>
<td>Maximal search distance $d$ should be used</td>
</tr>
</tbody>
</table>
*optionMappingMATFEMSettings
Enable special handling of LSDyna history variables when mapping a MATFEM GenYld - CrachFEM result file
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disabled (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

*optionReaderDynaMAT249
Active automatic transformation of ply orientation for history variables of LSDyna MAT_249
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disabled (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

*optionWriterDynaOutput
Define the mode output files for LSDyna are written
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Single and multiple file are written (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Single file only is written</td>
</tr>
<tr>
<td>-1</td>
<td>Multiple files only are written</td>
</tr>
</tbody>
</table>

*optionWriterDynaMAT58FiberMode
Automatic transformation of ply orientation into MAT_58
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Align element coordinate system using fiber 1 (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Align element coordinate system congruent bisectrix</td>
</tr>
</tbody>
</table>

*optionWriterDynaOrientationToHistory
Define how variable ORIENTATION_TENSOR from molding simulation is handled on LSDyna file export
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disabled (Default), ELEMENT_SOLID_ORTHO is written from ORIENTATION_TENSOR</td>
</tr>
<tr>
<td>1</td>
<td>Enabled, generate MATFEM GenYld - CrachFEM history variables</td>
</tr>
</tbody>
</table>

*optionDiableStress
Special flag to disable quantity STRESS for mapping
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Map quantity STRESS (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Disable quantity STRESS for mapping</td>
</tr>
</tbody>
</table>

*optionMappingQuantity
Define a list of quantities to map from source to target
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Map all quantities (Default)</td>
</tr>
<tr>
<td>#N &quot;Quant_1&quot; ...</td>
<td>Map only the N quantities defined by named list</td>
</tr>
</tbody>
</table>

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 serveral *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, DC3D5, C3D5, DC3D6, C3D6, DC3D8, C3D8, DC3D10, C3D10
4.1.3 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PLASTIC STRAIN</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>ORIENTATION</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>PLY VECTOR</td>
<td>-</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Boundary Condition Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>*NODAL THICKNESS</td>
</tr>
<tr>
<td></td>
<td>*DISTRIBUTION,LOCATION=ELEMENT,NAME=SHELL_THICKNESS_DIST</td>
</tr>
<tr>
<td>STRESS</td>
<td>*INITIAL CONDITIONS,TYPE=STRESS</td>
</tr>
<tr>
<td>PLASTIC STRAIN</td>
<td>*INITIAL CONDITIONS,TYPE=HARDENING</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>*DLOAD</td>
</tr>
<tr>
<td>ORIENTATION</td>
<td>*ORIENTATION</td>
</tr>
<tr>
<td>PLY VECTOR</td>
<td>*DISTRIBUTION TABLE using COORD3D, *DISTRIBUTION</td>
</tr>
<tr>
<td></td>
<td>*SHELL SECTION, , COMPOSITE</td>
</tr>
<tr>
<td></td>
<td>*DISTRIBUTION TABLE, *DISTRIBUTION and *ORIENTATION</td>
</tr>
</tbody>
</table>

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 Mechanical APDL

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) SHELL181.

Solids

Tetrahedrons (4 & 10 nodes), hexahedrons (8 & 20 nodes), pyramid (5 nodes) and prisms (6 nodes) are supported for reading and writing. Detailed list of supported ANSYS elements SOLID5, SOLID45, SOLID46, SOLID62, SOLID64, SOLID65, SOLID69, SOLID70, SOLID87, SOLID92, SOLID96, SOLID97, SOLID98, SOLID123, SOLID127, SOLID148, SOLID164, SOLID168, SOLID185, SOLID186, SOLID187, SOLID227, SOLID232, SOLID237 and SOLID285.
4.2.3 Transferable Quantities

Quantities THICKNESS, TEMPERATURE, STRESS, STRAIN, PLASTIC STRAIN, PRESSURE and ORIENTATION TENSOR are supported for mapping and writing. Boundary condition type INISTATE use default coordinate system identifier 0.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Boundary Condition Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>Tabular $Nodeid $Value</td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>BF, ,TEMP</td>
</tr>
<tr>
<td>STRESS</td>
<td>INISTATE, SET, DTYP, S</td>
</tr>
<tr>
<td>STRAIN</td>
<td>INISTATE, SET, DTYP, EPEL</td>
</tr>
<tr>
<td>PLASTIC STRAIN</td>
<td>INISTATE, SET, DTYP, EPPL</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>SFE,,,PRES</td>
</tr>
<tr>
<td>ORIENTATION TENSOR</td>
<td>Moldflow xml format, Digimat *.dof</td>
</tr>
<tr>
<td>WELD LINE</td>
<td>Element set or Digimat *.dof</td>
</tr>
<tr>
<td>DISPLACEMENTS</td>
<td>nblock coordinate with applied displacement</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE_1</td>
<td>*.NodeData_Thickness containing nodal coordinates and ids</td>
</tr>
<tr>
<td>FILE_2</td>
<td>*.ElementData_Thickness containing element topology and thickness</td>
</tr>
<tr>
<td>FILE_3</td>
<td>*.ElementData_Plastic_Strain containing element topology plastic strain</td>
</tr>
<tr>
<td>FILE_4</td>
<td>Result_File_Name</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>ORIENTATION_TENSOR</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>PLASTIC_STRAIN</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>
4.6 MSC Simufact

4.6.1 File Format

All MSC.Simufact products, Additive, Forming and Welding, offer a configurable result
export utilizing the Universal File format (*.unv), which is fully supported by MpCCI
Mapper.

4.6.2 Element Types

MpCCI Mapper supports reading of all element types available in MSC Simufact product.
The interface comprises reading solid, thick shell and shell elements.

4.6.3 Transferable Quantities

Both, nodal and element integration point results can be read from MSC.Simufact Uni-
versal files. The data in imported in SI units. The following list of output variables is
automatically handeled as MpCCI Mapper quantitites:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>PLASTIC STRAIN</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>DISPLACEMENTS</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>
4.7 Forge

4.7.1 File Format

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

4.7.2 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.

4.7.3 Transferable Quantities

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

4.8.1 File Format

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

4.8.2 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.

4.8.3 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
<th>Boundary Condition or Element Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>x</td>
<td>*ELEMENT_SHELL_THICKNESS</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRESS_SHELL</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRESS_SOLID</td>
</tr>
<tr>
<td>STRAIN</td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRAIN_SHELL</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRAIN_SOLID</td>
</tr>
<tr>
<td>PLASTIC_STRAIN</td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRESS_SHELL</td>
</tr>
<tr>
<td>HISTORY Variables</td>
<td>x</td>
<td>x</td>
<td>*INITIAL_STRESS_SHELL</td>
</tr>
<tr>
<td>ORIENTATION_TENSOR</td>
<td>-</td>
<td>x</td>
<td>*INITIAL_STRESS_SHELL</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>*ELEMENT_SOLID_ORTHO</td>
</tr>
<tr>
<td>ELEMENT_SHELL_BETA</td>
<td>x</td>
<td>x</td>
<td>*ELEMENT_SHELL_THICKNESS</td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>-</td>
<td>x</td>
<td>*INITIAL_TEMPERATURE_NODE</td>
</tr>
</tbody>
</table>

Table 4.3: MpCCI Mapper quantities in LSDyna solver format

4.8.4 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 quantities:

<table>
<thead>
<tr>
<th>Quantity Name</th>
<th>HISTORY Variable</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLY_VECTOR_1</td>
<td>POSTV dependent</td>
<td>Direction 1st fiber (global coordinates)</td>
</tr>
<tr>
<td>PLY_VECTOR_2</td>
<td>POSTV dependent</td>
<td>Direction 2nd fiber (global coordinates)</td>
</tr>
</tbody>
</table>

Table 4.4: MpCCI Mapper quantities for composites.
4.8.5 Injection Molding Features

The fiber ORIENTATION_TENSOR result shows the probability of fiber alignment in the specified principal direction at the end of an 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_SOLIDORTHO card. For material #1 a *MAT_ELASTIC template and for material 2-15 a *MAT_ANISOTROPIC_ELASTIC template is written to output files.

<table>
<thead>
<tr>
<th>Id</th>
<th>Material Title</th>
<th>SECTION_SOLID_TITLE</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MATERIAL_ISOORTHOTROPIC</td>
<td>PART_ISOORTHOTROPIC</td>
<td>a11 = a22 = a33 = 0.33</td>
</tr>
<tr>
<td>2</td>
<td>MATERIAL_0.4,0.3,0.3</td>
<td>PART_0.4,0.3,0.3</td>
<td>a11 = 0.4, a22 = 0.3, a33 = 0.3</td>
</tr>
<tr>
<td>3</td>
<td>MATERIAL_0.4,0.4,0.2</td>
<td>PART_0.4,0.4,0.2</td>
<td>a11 = 0.4, a22 = 0.4, a33 = 0.3</td>
</tr>
<tr>
<td>4</td>
<td>MATERIAL_0.5,0.3,0.2</td>
<td>PART_0.5,0.3,0.2</td>
<td>a11 = 0.5, a22 = 0.3, a33 = 0.2</td>
</tr>
<tr>
<td>5</td>
<td>MATERIAL_0.5,0.4,0.1</td>
<td>PART_0.5,0.4,0.1</td>
<td>a11 = 0.5, a22 = 0.4, a33 = 0.1</td>
</tr>
<tr>
<td>6</td>
<td>MATERIAL_0.5,0.5,0.0</td>
<td>PART_0.5,0.5,0.0</td>
<td>a11 = 0.5, a22 = 0.5, a33 = 0.0</td>
</tr>
<tr>
<td>7</td>
<td>MATERIAL_0.6,0.2,0.2</td>
<td>PART_0.6,0.2,0.2</td>
<td>a11 = 0.6, a22 = 0.2, a33 = 0.2</td>
</tr>
<tr>
<td>8</td>
<td>MATERIAL_0.6,0.3,0.1</td>
<td>PART_0.6,0.3,0.1</td>
<td>a11 = 0.6, a22 = 0.3, a33 = 0.1</td>
</tr>
<tr>
<td>9</td>
<td>MATERIAL_0.6,0.4,0.0</td>
<td>PART_0.6,0.4,0.0</td>
<td>a11 = 0.6, a22 = 0.4, a33 = 0.0</td>
</tr>
<tr>
<td>10</td>
<td>MATERIAL_0.7,0.2,0.1</td>
<td>PART_0.7,0.2,0.1</td>
<td>a11 = 0.7, a22 = 0.2, a33 = 0.1</td>
</tr>
<tr>
<td>11</td>
<td>MATERIAL_0.7,0.3,0.0</td>
<td>PART_0.7,0.3,0.0</td>
<td>a11 = 0.7, a22 = 0.3, a33 = 0.0</td>
</tr>
<tr>
<td>12</td>
<td>MATERIAL_0.8,0.1,0.1</td>
<td>PART_0.8,0.1,0.1</td>
<td>a11 = 0.8, a22 = 0.1, a33 = 0.1</td>
</tr>
<tr>
<td>13</td>
<td>MATERIAL_0.8,0.2,0.0</td>
<td>PART_0.8,0.2,0.0</td>
<td>a11 = 0.8, a22 = 0.2, a33 = 0.0</td>
</tr>
<tr>
<td>14</td>
<td>MATERIAL_0.9,0.1,0.0</td>
<td>PART_0.9,0.1,0.0</td>
<td>a11 = 0.9, a22 = 0.1, a33 = 0.0</td>
</tr>
<tr>
<td>15</td>
<td>MATERIAL_1.0,0.0,0.0</td>
<td>PART_1.0,0.0,0.0</td>
<td>a11 = 1.0, a22 = 0.0, a33 = 0.0</td>
</tr>
</tbody>
</table>

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

4.8.6 Output Files

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.key</td>
<td>File with model *NODE definition</td>
</tr>
<tr>
<td>*.key.thick</td>
<td>Include statements for *.key.thick and *.key.stressstrain</td>
</tr>
<tr>
<td>*.key.stressstrain</td>
<td>Model *ELEMENT definition with THICKNESS</td>
</tr>
<tr>
<td>*.complete.key</td>
<td>Combinend <em>.key,</em>.key.thick and *.key.stressstrain output</td>
</tr>
<tr>
<td>*.with_displacements.key</td>
<td>Optional file containing morphed coordinates when mapping solid to shell</td>
</tr>
</tbody>
</table>

Table 4.6: Output files generated from MpCCI Mapper

History variables

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 which keyword “MapperHistoryVariableConfiguration” the configuration is given in tabular format.

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

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 4.2 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.](image)

The same output can be generated using short notation

```
MapperHistoryVariableConfiguration
2 4 1.5
4 2
```
If the new history id is set to “0”, the variable is set to zero for all integration points and thus treated as empty.

Figure 4.2: History variable permutation imported in writer settings dialog.

Element formulation and integration scheme

If the LSDyna model is used as target model, MpCCI Mapper supports a check and redefinition of shell element formulation and through thickness integration rule (and number of points though thickness) for each shell section within the model. Using right-click option on target model name in “Target Models” tree 3.4.1 then selecting “Overwrite Integration Rule” an integration types dialog opens 4.3. Here, a default shell section integration formula is given as well for each shell section within the model an additional data row to change element formulation (reduced of fully integrated) as well as the number of integration points though thickness.
Note: If element formulation of the number of integration points though thickness is changed, all defined quantities of the model will be deleted. To obtain data on specified element formulation / number of integration points please re-run mapping step.
4.9 Marc / COPRA FEA

4.9.1 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.

4.9.2 Element Types

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

Table 4.7: Supported MSC Marc element types

4.9.3 Transferable Quantities

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMPERATURE</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>STRAIN</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>PLASTIC STRAIN</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>
4.10 Moldflow

4.10.1 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.

4.10.2 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>ORIENTATION_TENSOR</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>

4.11 Nastran

4.11.1 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.

4.11.2 Element Types

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

4.11.3 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>STRESS</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>PLASTIC_STRAIN</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Boundary Condition Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>element description card format</td>
</tr>
<tr>
<td>STRESS</td>
<td>ISTRSSH for SOL 700 only</td>
</tr>
<tr>
<td>PLASTIC_STRAIN</td>
<td>ISTRSSH for SOL 700 only</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>PLOAD2</td>
</tr>
</tbody>
</table>
4.12 PAM

4.12.1 File Format
MpCCI Mapper supports the following native PAM solver ASCII formats:
- PAMStamp mapping files (*M00-*M99)
- PAMCrash input deck (*.pc, *.ps)

4.12.2 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.

4.12.3 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
<th>Boundary Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>x</td>
<td>THIC</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>x</td>
<td>STRS</td>
</tr>
<tr>
<td>STRAIN</td>
<td>x</td>
<td>x</td>
<td>STRN</td>
</tr>
<tr>
<td>PLASTIC, STRAIN</td>
<td>x</td>
<td>x</td>
<td>PLAS</td>
</tr>
<tr>
<td>PRESSURE</td>
<td>-</td>
<td>x</td>
<td>PREFA</td>
</tr>
<tr>
<td>HV__</td>
<td>x</td>
<td>x</td>
<td>HV__</td>
</tr>
</tbody>
</table>

4.13 Radioss

4.13.1 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

4.13.2 Element Types
Fully integrated and reduced integrated triangles and quads are supported.

4.13.3 Transferable Quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICKNESS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>STRESS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PLASTIC, STRAIN</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
4.14 Sysweld

4.14.1 File Format

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.asc</td>
<td>Model information file</td>
</tr>
<tr>
<td>*_trans.asc</td>
<td>Mechanical or thermal result file. Stresses, strain, plastic strain and young modul is extracted. The effectiv plastic strain is extracted from plastic strain.</td>
</tr>
<tr>
<td>*_phase.asc</td>
<td>Defining names for the phases.</td>
</tr>
</tbody>
</table>

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

```
SYSWELD_PHASE_DEFINITION
BEGIN_PHASEDEF
    1 "MARTENSIT"
    2 "AUSTENIT"
    5 "SOMEOTHERUSERDEFINEDPHASE"
END_PHASEDEF
```

Output The files except 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:

```plaintext
# Ausgangsblechdicke: 1.00mm
```

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>u</td>
<td>v</td>
<td>x_coord</td>
<td>y_coord</td>
<td>z_coord</td>
<td>phi_1</td>
<td>phi_2</td>
<td>phi_3</td>
<td>%thickness reduction</td>
</tr>
</tbody>
</table>

Table 5.1: Order of Variables required for MpCCI Mapper

5.2 Atos

5.2.1 File Format

The 'Gesellschaft für 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](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

Either 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

<table>
<thead>
<tr>
<th>VTK</th>
<th>MpCCI Mapper</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTK_TRIANGLE</td>
<td>TRIAN</td>
<td>3-node shell element</td>
</tr>
<tr>
<td>VTK_QUAD</td>
<td>QUAD</td>
<td>4-node shell element</td>
</tr>
<tr>
<td>VTK_QUADRATIC_TRIANGLE</td>
<td>TRIAN6</td>
<td>6-node shell element</td>
</tr>
<tr>
<td>VTK_QUADRATIC_QUAD</td>
<td>QUAD8</td>
<td>8-node shell element</td>
</tr>
<tr>
<td>VTK_TETRA</td>
<td>TETRA</td>
<td>4-node volume element</td>
</tr>
<tr>
<td>VTK_HEXAHEDRON</td>
<td>HEXA</td>
<td>8-node volume element</td>
</tr>
<tr>
<td>VTK_WEDGE</td>
<td>WEDGE</td>
<td>6-node volume element</td>
</tr>
<tr>
<td>VTK_QUADRATIC_TETRA</td>
<td>TETRA10</td>
<td>10-node volume element</td>
</tr>
<tr>
<td>VTK_QUADRATIC_HEXAHEDRON</td>
<td>HEXA20</td>
<td>20-node volume element</td>
</tr>
</tbody>
</table>

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 THE COPYRIGHT HOLDERS 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 Tilera 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.

6.7 pugixml

This software is based on pugixml library (https://pugixml.org). pugixml is Copyright (C) 2006-2019 Arseny Kapoulkine.