IV Getting Started
# IV Getting Started – Contents

## 1 Multi-Physics Computation with MpCCI

1.1 Multi-Physics ................................................. 4
1.2 Solution of Coupled Problems .......................... 4
1.3 Code Coupling with MpCCI ............................... 5

## 2 Setting up a Coupled Simulation

2.1 A Simple Example ........................................... 7
2.2 Model Preparation ......................................... 7
   2.2.1 CFD Model ............................................ 8
   2.2.2 FE Model ............................................ 8
2.3 Starting the MpCCI GUI .................................... 9
2.4 Models Step – Choosing Codes and Model Files .......... 9
2.5 Coupling Step – Definition of Coupling Regions and Quantities ................................. 11
   2.5.1 Define Coupling Regions ............................. 12
   2.5.2 Select the Components for each Interconnected Code ........................................ 12
   2.5.3 Specify the Quantities which will be Exchanged ......................................................... 12
2.6 Monitors Step – Definition of Quantities for Monitoring .................................................. 13
2.7 Edit Step – Further Coupling Options .................... 14
2.8 Go Step – Starting Server and Codes ..................... 15
   2.8.1 Configuring the Initial Exchange Mode .................. 16
   2.8.2 Setting Option Parameters ............................ 18
   2.8.3 Starting the Coupled Simulation .................... 18
   2.8.4 Interrupting the Computation ........................ 22

## 3 Checking the Results

3.1 The MpCCI Visualizer ...................................... 24
3.2 Post-Processing .............................................. 24
1 Multi-Physics Computation with MpCCI

1.1 Multi-Physics

The purpose of MpCCI is to perform multi-physics simulations. The systems under consideration are known as coupled systems. A coupled system consists of two or more distinct systems. Each system is governed by a characteristic set of differential equations, but both systems share some variables and cannot be solved separately (see also Zienkiewicz and Taylor [2000] for a more precise definition).

\[
\begin{align*}
  & \text{domain A} \quad \text{domain B} \\
  & f(a,b,c,d) = 0 \\
  & g(c,d,e,f) = 0 \\
  & \text{coupling region}
\end{align*}
\]

Figure 1: Coupled system: Two domains and a coupling region. Both systems share the variables c and d in the coupling region

Figure 1 shows two coupled systems. In the coupling region, both systems share some variables and the governing equations of both systems must be solved. Depending on the dimension of the coupling region, surface coupling and volume coupling are distinguished. The shared variables (c and d in Figure 1) are called coupling quantities.

Typical multi-physics simulations are:

**Fluid-Structure Interaction (FSI):**
- First system: Fluid flow (Navier-Stokes equations)
- Second system: Solid mechanics (equilibrium)
- Quantities: Pressure (1 → 2), deformation (2 → 1)

**Thermomechanical coupling**
- First system: Solid mechanics (equilibrium)
- Second system: Heat conduction (Fourier’s law)
- Quantities: Temperature (2 → 1), deformation (1 → 2)

**Electrothermal coupling**
- First system: Electrical conduction (Maxwell’s equations)
- Second system: Heat conduction (Fourier’s law)
- Quantities: Temperature/electric conductivity (2 → 1), power loss/Joule heat (1 → 2)

1.2 Solution of Coupled Problems

To find a solution for a coupled problem, all governing equations, which can be combined in a large system, must be solved. The solution in this way is called strong coupling. However, solving a system with strong coupling is often difficult as different approaches are necessary to solve the sub-problems.
An alternative approach is through *weak coupling*. Here each problem is solved separately and some variables are exchanged and inserted into the equations of the other problem. This procedure usually yields a less exact solution compared to *strong coupling*. The advantages of the weak coupling are that the sub-problems can be solved faster than the complete system and that specialized solvers can be used for each.

The staggered method is sketched in Figure 2, which is one of the weak coupling approaches of MpCCI: Code A computes one step, sends data to code B, which then computes one step and sends data back to code A and so on. In addition to the staggered approach, MpCCI supports parallel execution of both codes. The selection of the coupling algorithm is described in \( \triangleright \) 2.8 Go Step – Starting Server and Codes \( \triangleright \).

An important issue is the data exchange. The quantities must be transferred from one code to the other. This process can be divided into two parts:

**Association:** Each point and/or element of the components of a coupling region is linked to a partner in the other system. The process of finding partners is also called *neighborhood search*, see \( \triangleright \) V-3.3 Data Exchange \( \triangleright \).

**Interpolation:** The quantities must be transferred to the associated partner on the other mesh. In this process, the different mesh geometries, data distributions and the conservation of fluxes must be considered.

MpCCI fully supports the data exchange between non-conforming meshes, i.e. the meshes of each subsystem can be defined to optimize the solution of the subsystem.

### 1.3 Code Coupling with MpCCI

Running a co-simulation with MpCCI requires the following steps:

**Preparation of Model Files.** Before starting a co-simulation, each domain must be modeled separately, i.e. a model file must be created for each simulation code. The models must contain a definition of the coupling regions. The preparation of a model file depends on the simulation codes, a detailed description is given in the corresponding sections of the Codes Manual.
**Definition of the Coupling Process.** Simulation codes and corresponding model files must be selected. The coupled regions, quantities and a coupling algorithm must be selected, further coupling options can be given. This step is completely supported by the MpCCI GUI.

**Running the Co-simulation.** After starting the MpCCI server, both coupled codes are started. Each code computes its part of the problem while MpCCI controls the quantity exchange.

**Post-Processing.** After the co-simulation, the results can be analyzed with the post-processing tools of each simulation code, with the MpCCI Visualizer or with general-purpose post-processing tools.

The application of MpCCI requires a good knowledge of the employed simulation codes. Therefore, it is recommended to use those codes for a co-simulation the user has already some experience with.

The data exchange between two simulation codes requires a “code adapter”, which constitutes a “code plug-in” for data transfer. Therefore only codes which are supported by MpCCI can be used. It is also possible to develop special code adapters, more details are given in the Programmers Guide.

An overview of the complete co-simulation process is given in Figure 3.
2 Setting up a Coupled Simulation

2.1 A Simple Example

Let us look at a simple example to clarify the approach, which was described in the preceding chapter. Figure 1 shows a model of a valve which consists of a flexible flap which can be open or closed depending on the direction of the flow. The purpose of the simulation is to investigate the behavior of the valve depending on inlet and outlet pressure.

The computation of the pressure distribution and the flow rates is achieved using Computational Fluid Dynamics (CFD), whereas the deformation of the flap can be computed applying numerical structural mechanics via a Finite Element (FE) code. Each domain, has a significant influence on the other, thus the problems cannot be solved separately. The CFD simulation requires the deformation of the flap as a boundary condition, whereas the FE simulation requires the fluid pressure as external load. Thus during a co-simulation, these quantities must be exchanged as shown in Figure 2.

2.2 Model Preparation

Before starting the coupled simulation, the models must be prepared in each code. In our example, this requires a model of the flow region and a model of the flap structure.

Both models are created in the undeformed condition, i.e. the flap is straight. The computation of such a problem requires a CFD-code which can handle moving/deformable meshes.

It is recommended that you keep your FE and CFD model files in separate directories (see Figure 3) because of the following reasons:

- Clear storage and maintenance of simulation data.
- Ensuring that the codes do not overwrite files of other codes, when identical file names are used.
- Simplification of the porting of the analysis when running the simulation on different platforms.
2.2.1 CFD Model

To model the fluid domain, define the geometry in the CFD pre-processor and create a mesh. Figure 4 shows a possible mesh for the fluid domain. (Please note that the displayed mesh is rather coarse and would probably not yield good results).

All boundary conditions must be defined as well. They comprise the flap surface, walls, inlet and outlet in Figure 4. Also for the coupling region boundary conditions may be set - this depends on the CFD code applied, please check the Codes Manual for details.

It is recommended to run the fluid problem on its own before starting a coupled simulation to be sure that the model setup is correct. In the present example the coupling region could be defined as a rigid wall for this purpose. If the model is not appropriate for computations with the CFD code alone, it is most likely that a co-simulation will either not run.

2.2.2 FE Model

Compared to the CFD model which covers the fluid domain, the FE model covers the flap itself. Figure 5 shows a mesh of the flap with quadrilateral elements. The element sizes do not need to correspond to those of the fluid domain in any way as MpCCI can handle non-conforming meshes.

The flap is connected to the top wall, i.e. the top nodes of the mesh must be fixed. The surface of the flap must be defined as a boundary to which the pressure load can be applied in the co-simulation. Although the system will not show any deformation without an external load, it is recommended to check the validity of the mesh and boundary conditions (i.e. the fixed nodes here) by running the FE problem alone and/or by performing a problem check if it is available in the FE code. To obtain a deformation you can replace the fluid forces by a similar load.
2.3 Starting the MpCCI GUI

The MpCCI GUI is started by executing the command `mpcci gui` from a shell console.

The MpCCI GUI guides you through the steps to interconnect the FE and CFD models and to run the coupled simulation.

2.4 Models Step – Choosing Codes and Model Files

In the Models Step, the following steps should be accomplished:

- Select the analysis codes to couple.
- Specify their model files.
- Set some code specific parameters if desired.
- Scan the model files to determine the potential coupling regions.

For each code, `Code_1` and `Code_2` (Figure 6), a field for selecting and configuring the analysis code is provided in the MpCCI GUI.

For the CFD-Code: Select the desired CFD-Code from the pull-down menu providing the available codes. After your selection the parameters and settings of the picked solver are shown underneath (see Figure 6).

1. Select the CFD model file by clicking on the [Browse] button in order to choose the file via the file browser.
2. You may also configure some additional parameters like the CFD version or release.
3. Scan the model file by clicking on the [Start Scanner] button.

For the FE-Code: Select the desired FE-Code from the list provided by the pull-down menu. After your selection the corresponding parameters and settings are shown underneath (see Figure 7).

1. Select the FE model file via the file browser by clicking on the [Browse] button.
2. Select the unit system.
3. If the unit system is set to variable an additional parameter will be shown where you can select the grid length unit.
4. In addition you may change the settings for the FE release or the scan method.

5. Scan the model file by clicking on the `Start Scanner` button.

For each scanned model file the status is assigned to the corresponding analysis code:

![DONE] if the extraction of the interface regions was successful.
If the scanner encountered problems during the scan which may depend on the parameters set for the analysis code.

By clicking on the status button, you get the information that has been extracted from the model file. If the scanning has failed the error message from the scanner is displayed in a pop-up window.

After performing a successful scan of all model files you may continue with the next step of the coupled simulation setup by clicking on the Next button.

### 2.5 Coupling Step – Definition of Coupling Regions and Quantities

In the Coupling Step the following steps need to be done:

- Define the coupling regions.
- Select the components for each interconnected code.
- Specify the quantities which will be exchanged.

The components are automatically sorted by their element type. There are basically two types of components:

**Global Variables** These components are data structures that are not related to the CFD or FEM grids. They contain global quantities like time or time step size. These components can be found under the Global element type label.

**Mesh Based Element Components** These components comprise collections of mesh based elements and can be found under the Mesh label. They contain model parts and the related grid based quantities, e.g. nodal positions, heat values or forces. To build up the interconnection, elements should be gathered that are part of the coupling region. Therefore the components of the codes are collected in lists of zero-dimensional integration point elements, zero-dimensional point elements, one-dimensional line elements, two-dimensional face elements and three-dimensional volume elements (see Figure 8).

![Figure 8: Integration Point, Point, Line, Face and Volume element labels](image)

To navigate between the two component types you have to click on the corresponding tab: Global or Mesh. Only active element types are offered (see Figure 9).

In our example only the Mesh label is shown and already selected.

Here in the code component list only Face elements exist, so only this one list is shown.

For each component type the MpCCI GUI window is divided into three fields:

- **Regions** with a list of the created coupling regions.
- **Components** with the lists of code components which may be selected into the coupled components lists, which define the coupling region.
- **Quantities** with a list of available quantities which may be selected and configured to be transferred from one application to the other on the selected coupling region.
2.5.1 Define Coupling Regions

In the field “Regions” there is already the default region Mesh_1 defined. To change its name use Rename from the menu which pops up when you click with the right mouse button onto the region name. To add further coupling regions use the Add button at the top of the regions list (Figure 10).

2.5.2 Select the Components for each Interconnected Code

In the Components part for each code a list with available components for that particular code is shown. According to the valve example with the flexible flap this will imply: For CFD-code select the component flap-surface by double clicking, using drag & drop or using Add to Region Set from the pop-up menu (see Figure 11).

For FE-code select the component outside in the same way as mentioned above.

2.5.3 Specify the Quantities which will be Exchanged

In the quantities list, select the quantities to be transferred by clicking on the checkbox near the name of the quantity (Figure 12). In MpCCI the names of the quantities are standardized. In our example we want
to transfer pressure and deformation. In MpCCI the pressure is handled by the quantity RelWallForce and the deformation by NPosition which stands for “nodal position”. For further information on the meaning of quantities see the Quantity Reference in the Appendix and \textgreater{} V-3.1.1 Physical Domains \textless{}. For our example select and configure the following quantities:

- **RelWallForce** for exchanging the pressure (see Figure 12). Sender is CFD-code. Its pressure unit is not editable because the unit system for the CFD-code is fixed to the “SI” unit system. On the other hand you may select the appropriate unit for the FE-code because of the “variable” unit system setting in the Models Step (see Figure 7).

- **NPosition** for exchanging the deformation (see Figure 13). Sender is FE-code. The unit for the deformation may be set as described above.

For further information have a look at \textgreater{} V-4.4 Coupling Step \textless{}.

The definition of the coupling interface is completed, please click on the \textbf{Next} button to continue.

### 2.6 Monitors Step – Definition of Quantities for Monitoring

In the Monitors Step you can define some quantities for specific code components to be monitored. For the time being we won’t apply monitors. Please click on the \textbf{Next} button to continue.
2.7 Edit Step – Further Coupling Options

In the Edit Step you may modify some MpCCI control parameters like

- online monitoring setting,
- parameters for search algorithm,
- switches controlling the output level for debugging or
- tracefile file format.

An overview of all settings is given in V-4.6 Edit Step.<

The MpCCI GUI provides default values for all settings which can be modified.
The Edit Step window displays a tree of parameters on the left. On the right side the corresponding parameters are shown in a table as depicted in Figure 14. This table provides information such as

- the parameter name displayed with an orange background,
- the editable parameter value displayed with a yellow background,
- and a parameter description.

To select a parameter click on the parameter name in the parameters tree. That followed edit the value by clicking the cell with the parameter and select or type in the desired value for this parameter.

For most cases the default values are appropriate, but for our example you may activate additional options for the CCVX tracefile format (Output → TraceFile → Writers → Orphans) which is the option for writing orphans nodes information for the MpCCI Visualizer. Additionally you may edit the output level (Control → Level) which tells MpCCI how much output will be written during the simulation process.

Now click on the Next button to proceed to the start-up window for the coupled simulation.

### 2.8 Go Step – Starting Server and Codes

In the Go Step each application including the MpCCI coupling server is shown in its own frame. The frame for the codes is titled with the code’s name and the name of the host where they’re running (see Figure 15). Before starting the coupled simulation you have to configure the start-up of the applications. For the MpCCI coupling server you usually may retain the default values. For each analysis code you have to configure the initial exchange mode and set some option parameters if necessary.


2.8.1 Configuring the Initial Exchange Mode

In our coupled valve problem we consider that the pressure will initiate the deformation of the flap. Therefore the CFD-code will be the initiator and the FE-code in our example will request the pressure solution before starting its computation. After both codes have performed the initial data exchange they run the analysis computation until a coupled target time is reached. At this target time both analysis codes will exchange their solution quantities.

For the CFD-code, set the Initial quantities transfer mode to send as in Figure 16 by checking the Coupling configuration option which populates additional settings.

![Figure 16: CFD-code settings](image)
For the FE-code, set the initial quantities transfer mode to receive (see Figure 17) by checking the Coupling configuration option which populates additional settings.

Figure 17: FE-code settings
2 Setting up a Coupled Simulation

With this initial quantities transfer configuration for the CFD- and FE-code, the coupled simulation will use a parallel coupling algorithm (see Figure 18).

### 2.8.2 Setting Option Parameters

For the CFD-code select the Run parallel option. Now the parameters for running the CFD-code in parallel are added to the window as shown in Figure 16. Set the number of processes to be applied e. g. to 4. If no hosts are specified the CFD-code will run in parallel on the local machine.

For the FE-code no more options need to be set.

### 2.8.3 Starting the Coupled Simulation

Before starting a coupled simulation you have to establish a project by using the action Save As from the File menu. This is due to the fact that MpCCI needs the project file to generate its input file and to get information about the settings which were defined in the MpCCI GUI.

Server and each application have their own Start button. But only the Start button from the MpCCI server is enabled because the server has to be launched in advance. After the server has been started its Start button will be replaced by a running symbol, all settings will be locked and the Start buttons for the next applications will be enabled one after the other.

For our valve example the start-up procedure will be:

1. Click on Start to launch the coupling server. Now the server waits to get response from all client codes.
2. The Start button of the next application (CFD-code) is enabled. Now click on Start to launch the CFD analysis code.
3. After that the last application (FE-code) has to be launched by a click on its Start button.

Now the client codes are started and the coupling server starts the initialization phase as follows:

1. Initial handshaking: the CFD-code and the FE-code contact the server.

   Waiting for incoming connections on "47010@nemo"...
   [MpCCI License] mpcci_adapter_CFD: feature test...
   [MpCCI License] mpcci_clients: feature checkout...
   Waiting for incoming connections on "47010@nemo"...
   [MpCCI License] mpcci_adapter_FE: feature test...
   [MpCCI License] mpcci_clients: feature checkout...
   Loading code specific mesh and quantity settings for code "CFD":
   [...]

2. Exchange of the interface topology: The coupling server requests the interface topology from each client. The clients send their interface topology.

   Loading code specific mesh and quantity settings for code "CFD":
Code specific quantity properties:

"RelWallForce": Location(CODE), Default(0)
Send : Method(0), Index(0)
Receive : Method(0), Index(-1)

"NPosition": Location(CODE), Default(0)
Send : Method(0), Index(-1)
Receive : Method(0), Index(0)

Code specific parts properties:

"flap-surface": MeshId(1), PartId(1)
Send : "RelWallForce"
Receive: "NPosition"

Code specific mesh scale: 1

Code specific mesh transformation:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Det: 1

Loading code specific mesh and quantity settings for code "FE":

Code specific quantity properties:

"RelWallForce": Location(CODE), Default(0)
Send : Method(0), Index(-1)
Receive : Method(0), Index(0)

"NPosition": Location(CODE), Default(0)
Send : Method(0), Index(0)
Receive : Method(0), Index(-1)

Code specific parts properties:

"outside": MeshId(1), PartId(1)
Send : "NPosition"
Receive: "RelWallForce"

Code specific mesh scale: 1

Code specific mesh transformation:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Det: 1
3. MpCCI coupling server performs neighborhood searches and computes the mapping between CFD-code and FE-code meshes.

[...]

Loaded REL operator library "rel_ml-32d.so".

Loaded MAP operator library "map_ml-32d.so".

Code/Mesh/Part/Quantities relationships:

Code: "CFD", ID(1), nice(64), clients(1), type(Finite Volume).

Mesh: "CFD/MESH-1", mid(1)
  Coord system: 3D
  Mesh type: FACE
  Distances: [0.001 .. 0.00312839]
  Bounding box: [-0.00717758 .. 0.00717758]
    [-0.023 .. 0.025]
    [-0.023441 .. 0.023441]
  Domain size: 0.004896

Send: "RelWallForce"
  Dimension: 1
  Direction: SEND
  Location: cell
  Default: 0
  Buffers: 1

Recv: "NPosition"
  Dimension: 3
  Direction: RECV
  Location: vertex
  Default: 0
  Buffers: 1
  Source: "FE/MESH-1" -> rel_ml -> map_ml

Part: "CFD/MESH-1/flap-surface", pid(1)
  Coord system: 3D
  Mesh type: FACE
  NVertices: 997
  NCells: 1948
    Type1: TRIA3
    Ntypes: 1
  Total nodeids: 5844
  Total vertices: 5844
  Distances: [0.001 .. 0.00312839]
  Bounding box: [-0.00717758 .. 0.00717758]
    [-0.023 .. 0.025]
    [-0.023441 .. 0.023441]
  Domain size: 0.004896

Code: "FE", ID(0), nice(64), clients(1), type(Finite Element).
Mesh: "FE/MESH-1", mid(1)
  Coord system: 3D
  Mesh type: FACE
  Distances: [0.0005 .. 0.0033941]
  Bounding box: [-0.007178 .. 0.00717784]
                          [-0.023 .. 0.025]
                          [-0.0234407 .. 0.023441]
  Domain size: 0.00489598

Send: "NPosition"
  Dimension: 3
  Direction: SEND
  Location: node
  Default: 0
  Buffers: 1

Recv: "RelWallForce"
  Dimension: 1
  Direction: RECV
  Location: element
  Default: 0
  Buffers: 1
  Source: "CFD/MESH-1 --> rel_ml --> map_ml"

Part: "FE/MESH-1/outside", pid(1)
  Coord system: 3D
  Mesh type: FACE
  NNodes: 2805
  NElements: 920
    Type1: QUAD8
    Ntypes: 1
  Total nodeids: 7360
  Total vertices: 3680
  Distances: [0.0005 .. 0.0033941]
  Bounding box: [-0.007178 .. 0.00717784]
                          [-0.023 .. 0.025]
                          [-0.0234407 .. 0.023441]
  Domain size: 0.00489598

Neighborhood relationship: "FE/MESH-1" --> "CFD/MESH-1"
rel_ml::create_mapmesh(MESH="FE/MESH-1"): new mesh representation.
rel_ml::create_mapmesh(MESH="CFD/MESH-1"): new mesh representation.
rel_ml::execute(MAPLIB_REL=0x8111728): Configuration:
  nodetolerance=8.38e-05
  normaldistance=0.000419
  tangentialdistance=0.000419
  distance=0.000419
  precision=1e-06
  multiplicity=15.5513
  orphaninfo=true
Starting closePoints search loop...
rel_ml::execute(MAPLIB_REL=0x8111728)
Neighborhood relationship: "CFD/MESH-1" -> "FE/MESH-1"
rel_ml::create_mapmesh (MESH="CFD/MESH-1"): using existing mesh representation.
rel_ml::create_mapmesh (MESH="FE/MESH-1"): using existing mesh representation.
rel_ml::execute (MAPLIB_REL=0x825dd00): Configuration:
  nodetolerance=8.38e-05
  normaldistance=0.000419
  tangentialdistance=0.000419
  distance=0.000419
  precision=1e-06
  multiplicity=15.5513
  orphaninfo=true
Starting closePoints search loop...
rel_ml::execute (MAPLIB_REL=0x825dd00)
  -> 0.08 seconds CPU.

In our example the parallel coupling algorithm is used with send (CFD) and receive (FE) as initial exchange mode (see Figure 18 bottom). The CFD-code is represented by “code B” and the FE-code by “code A”. Before starting the computation the CFD-code sends the surface forces RelWallForce to the FE-code (1). Then both codes compute in parallel their solution quantities pressure and deformation until the coupled target time is reached (2). Now the solution quantities are exchanged (3). Possible coupling algorithms are described in detail in ⊿ V-3.4 Coupling Algorithms ⊿.

When an application exits normally, a window displays information collected from the application output.

2.8.4 Interrupting the Computation

While the applications are running, the Kill and Stop buttons at the bottom of the window are enabled and provide following functionalities:

- **Kill** to terminate all applications by sending a kill signal or by executing the "Killer" application module if available.

- **Stop** to terminate all applications by sending a stop signal or by executing the "Stopper" application module if available. The stop call may not immediately take effect, because it depends on the implementation of the stop procedure in the simulation code.

To modify parameter settings you have to terminate the applications by pressing the [Kill] button. Otherwise the editing of parameter values is disabled.
Figure 18: Initial exchange and resulting coupling algorithms for two codes with “exchange before solution”
3 Checking the Results

3.1 The MpCCI Visualizer

The MpCCI Visualizer is suitable for a quick check whether the coupling process was successful. The coupling region, orphaned nodes and exchanged quantities can be checked to ensure that the specified coupling has really occurred. Here only a short introduction is presented, a concise description of the MpCCI Visualizer is given in V-6 MpCCI Visualizer.

During the coupling process the MpCCI server writes a “tracefile”, i.e. a collection of the exchanged data (see Figure 3 and V-2.8 Go Step – Starting Server and Codes). To obtain a tracefile, the checkbox Use must be selected in the Edit Step for the selected Writers. The default name of the tracefile is "mpccirun-0000.ccvx", it can be changed in the Go Step of the MpCCI GUI by changing the Job name prefix for job files. This prefix is used to create a directory containing the tracefile. This directory begins with the prefix used and ends with the suffix of the type of writer used, e.g. mpccirun_ccvx.

After a simulation the MpCCI Visualizer can be started by selecting Tools → Visualizer from the MpCCI menu or by entering the command mpcci visualize. Tracefiles can even be opened with the MpCCI Visualizer before the completion of a computation.

Example:

```plaintext
mpcci visualize mpccirun_ccvx/mpccirun-000.ccvx
```

The visualizer starts with the results window, which is depicted in Figure 1. On the left side the data to display can be selected including the exchanged quantities by activating the Result Panel. In Figure 1 the quantity NPosition is additionally displayed as deformation in the window. Both the quantities which are sent, i.e. before the interpolation, and the quantities which are received are displayed. As sent and received data are normally defined at the same locations, both parts can be superimposed or separated by using additional viewports.

Transient analyses consist of several steps, the step number can be selected in the control window.

The MpCCI Visualizer can not replace a post-processing tool, as the tracefile only obtains information from MpCCI, which solely covers the coupling region. Information on other regions of the analysis may be available by creating additional monitors in the Monitor Step.

3.2 Post-Processing

During a coupled simulation both codes should write their results to appropriate files. The visualization of these results can be carried through with built-in tools of the simulation codes. Unfortunately the built-in post-processing tools solely allow the visualization of one part of the problem. In our example this means, that you can display the fluid properties with one tool and the structural properties with the other tool.

There is general post-processing software, which can read output data from different standardized formats and combine results from different files. Describing these tools is however beyond the scope of this manual.
Figure 1: Main window of the MpCCI Visualizer