

Release 2023 R1 Highlights

System Coupling



/ 2023 R1 Update Highlights

- PyAnsys/PySystemCoupling
- Aerodamping
- Automatic Interface Alignment
- Mapping Accuracy Improvements
- User Interface Improvements
- Other updates



System Coupling

PySystemCoupling

- Open-source Python module for access to System Coupling
- github.com/pyansys/pysystem-coupling
- Targeting first release at 2023 R1 RTP (January 2023)
- Focus on desktop deployment
 - Allows full breadth of System Coupling workflows to be executed from a Python console
 - Allows easy integration with other PyAnsys products
- Documentation
 - Automatically generated APIs reference
 - Examples via Sphinx gallery

The screenshot shows the top navigation bar of the PyAnsys documentation site with links for 'Getting Started', 'User Guide', 'API Reference', and 'Contributing'. Below the navigation bar is a search box labeled 'Search the docs ...'. The main content area is titled 'Getting Started' and contains introductory text about running PySystemCoupling, supported versions (2023 R1 or newer), and a link to visit Ansys for more information on getting a licensed copy.

The screenshot shows the 'solution_root' class page in the PyAnsys documentation. It features a search box and a table of commands. The class definition is shown as `class ansys.systemcoupling.core.adaptor.api.solution_root.solution_root(name: Optional[str] = None, parent=None)`. The table lists various commands with their summaries.

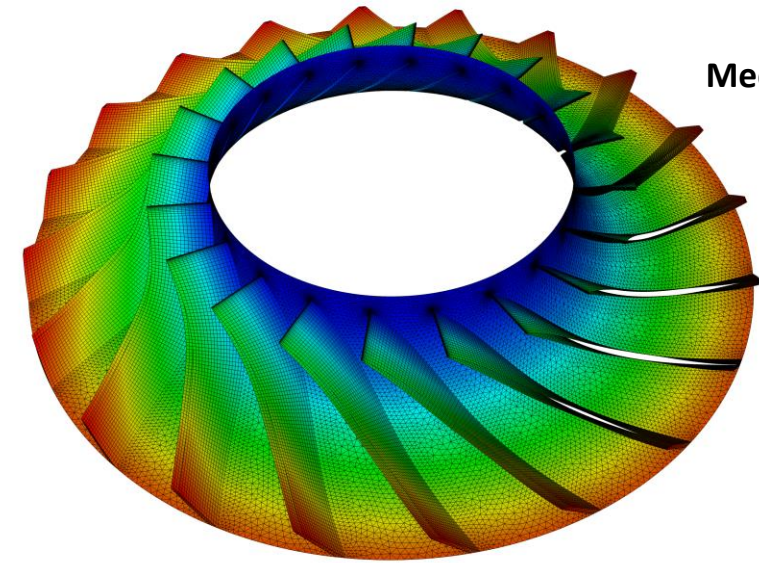
Command	Summary
start_participants	Important: This command will be deprecated. Consider adopting workflows
initialize	Interactive command that initializes a coupled analysis.
shutdown	Interactive command that shuts down a coupled analysis.
solve	Starts the participants (if necessary) and solves the coupled analysis. By:
step	Interactive command that initializes the analysis (if necessary) and
partition_participants	Provide a utility for setting the parallel algorithm, parallel partitioning
open_results_in_ensight	Allows for System Coupling results to be postprocessed in EnSight.
write_ensight	Write a file with mesh and results which can be loaded into EnSight for
create_restart_point	Interactive command that creates a restart point at the end of the
write_csv_chart_files	For each coupling interface, exports a CSV file containing chart data
get_machines	Get currently available machines.
interrupt	Interrupts a solve in progress.
abort	Aborts a solve in progress.

The screenshot shows the README.rst file for PySystemCoupling. It includes the project title, badges for PyAnsys, GitHub CI, passing, License MIT, code style black, and an 'Overview' section. The overview text states that the project provides Pythonic access to Ansys System Coupling, embedded within the Python ecosystem. It lists three key capabilities: launching System Coupling locally, accessing APIs for setup and solving, and full access to the data model via a Pythonic interface.

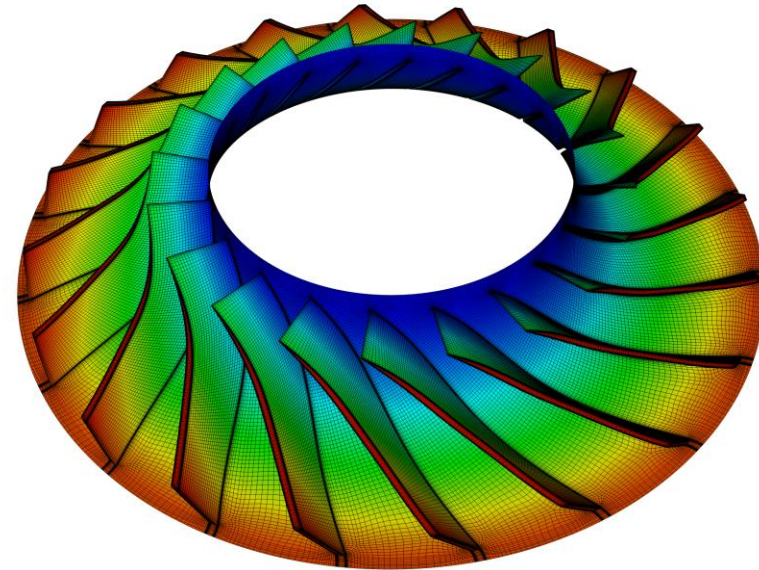
The screenshot shows the 'Examples' page in the PyAnsys documentation. It features a search box and a section titled 'Coupled Analysis Examples'. The text explains that end-to-end examples show how to use PySystemCoupling, including installing the `ansys-systemcoupling-core` package and running examples locally. Below the text is a gallery of two examples: 'Oscillating Plate tutorial case' and 'Parametric sweep example'. The gallery is generated by Sphinx-Gallery.

Aerodamping

- Enable mapping of complex mode shapes for cases with cyclic symmetry
- Multi-region support
 - For cases where blade surface, tip, hub, etc. are in separate regions
- Full access to System Coupling features for easier mapping
 - Instancing
 - Geometry transformations
 - Non-overlap extrapolation algorithm
 - Previewing mapping results in EnSight
 - Automatic interface alignment (beta)
 - Etc.



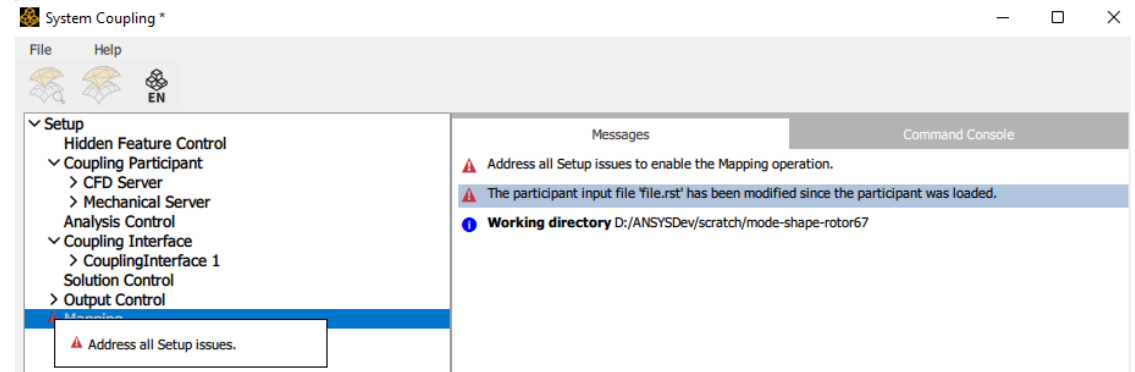
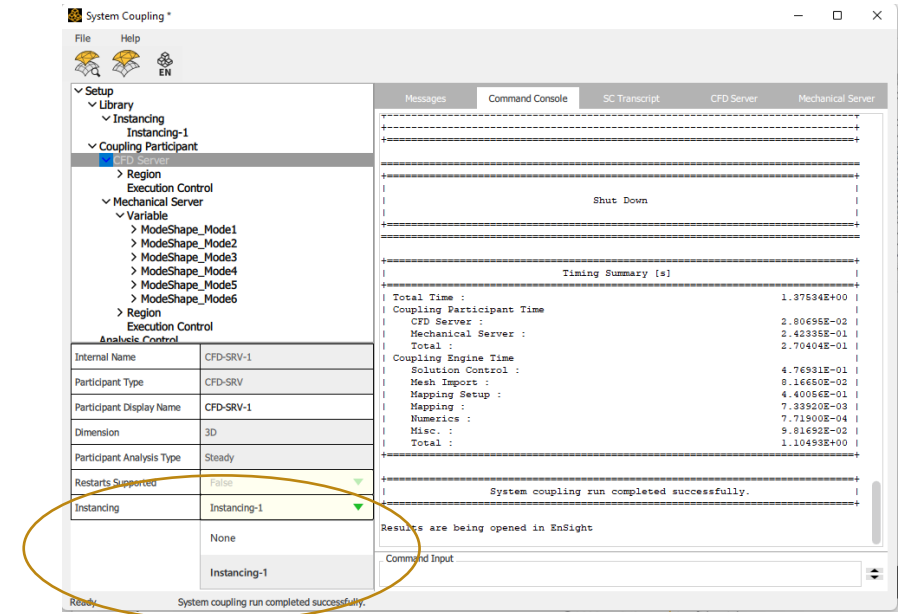
Source:
Mechanical modal
analysis



Target:
CFX analysis

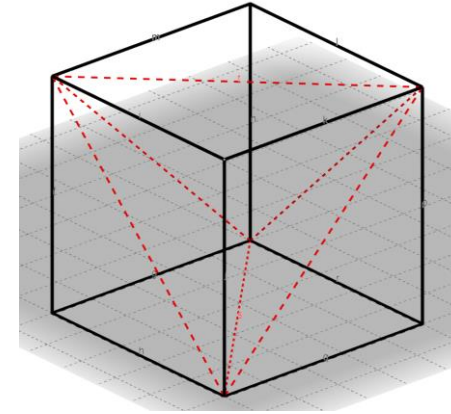
/ Aerodamping

- Specify CFD instancing from System Coupling GUI
- Detect input file changes if the underlying modal analysis was updated

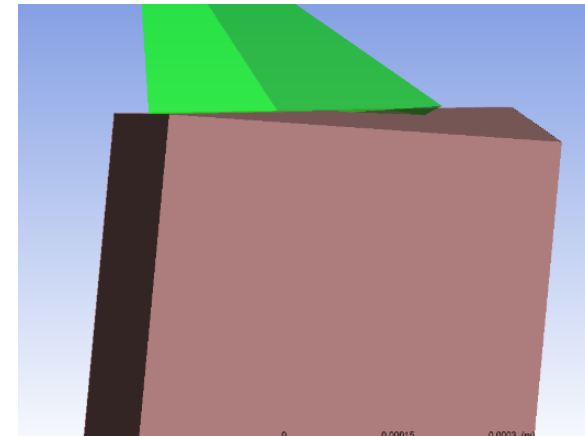


Mapping Accuracy Improvements

- Warped faces handling
- Volume mapper uses optimized cell splitting algorithm
 - Assumes all faces are planar
- Detect if a face is warped or not
- Different in 2023 R1
 - Turn off optimized cell splitting for cells with warped faces
 - Improves conservation
 - Turn on optimized cell splitting for high-order cells without warped faces
 - Dramatic speedup in mapping
 - 120 [s] -> 4 [s] for a test case



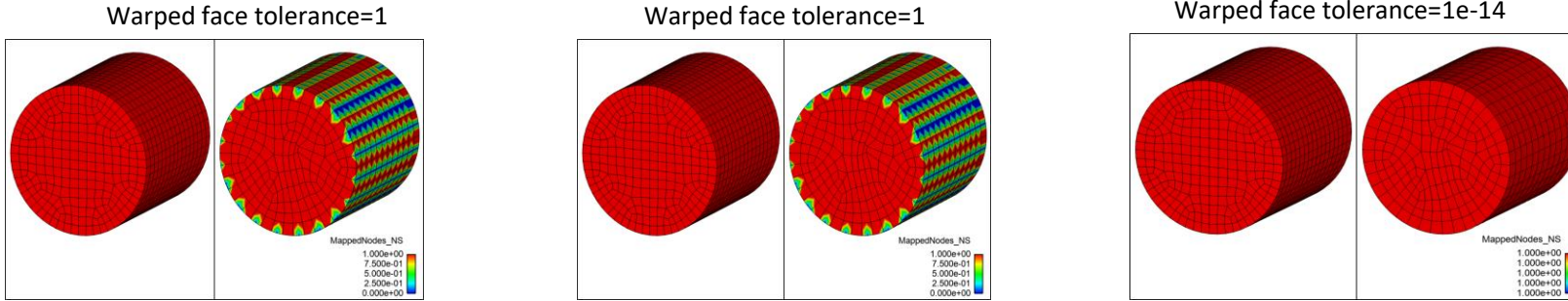
Optimized cell splitting assumes cells with no warped faces



Hexahedron element with warped faces renders optimized cell splitting inaccurate

Mapping Accuracy Improvements

- Warped mesh faces handling



Better mapping quality with warped faces detection turned on

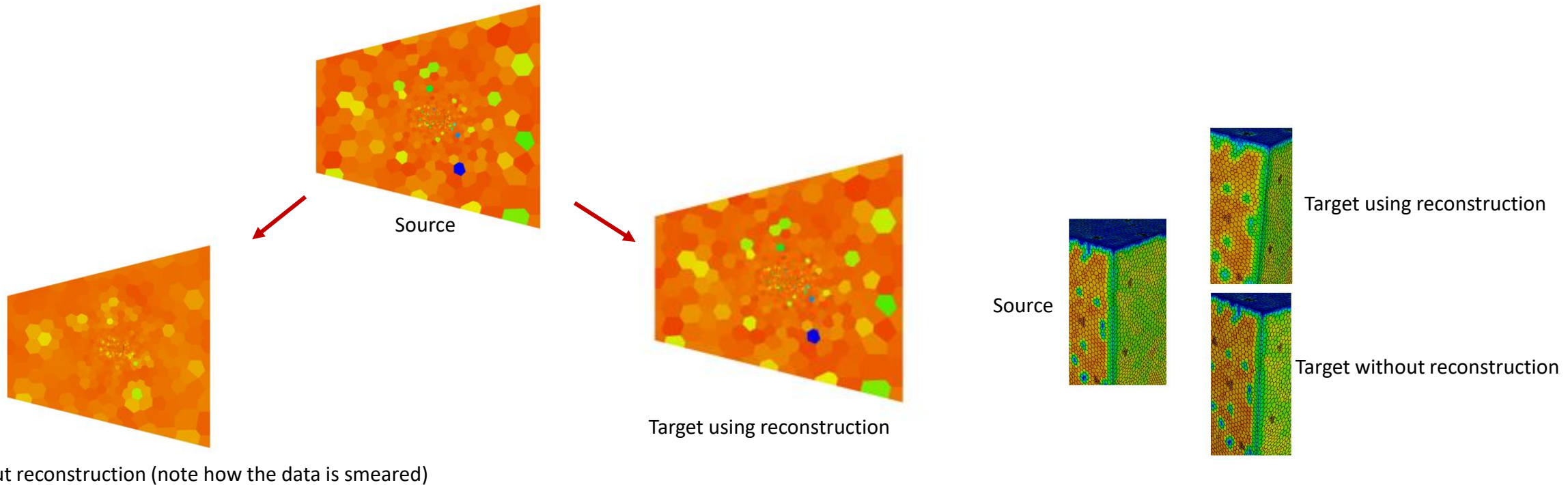
Mapping accuracy visualization with warped face tolerance 1,1e-6,1e-14 (from left to right)

MESH STATISTICS			MESH STATISTICS			MESH STATISTICS		
Participant: FLUENT-1			Participant: FLUENT-1			Participant: FLUENT-1		
Number of cell regions	1		Number of cell regions	1		Number of cell regions	1	
Number of cells	992		Number of cells	992		Number of cells	992	
Hexahedral	992		Hexahedral	992		Hexahedral	992	
Number of faces	3 278		Number of faces	3 278		Number of faces	3 278	
Volume (m3)	2.449e-02		Volume (m3)	2.449e-02		Volume (m3)	2.449e-02	
Bounding Box (m)			Bounding Box (m)			Bounding Box (m)		
Minimum	[-2.500e-01 0.000e+00 -2.500e-01]		Minimum	[-2.500e-01 0.000e+00 -2.500e-01]		Minimum	[-2.500e-01 0.000e+00 -2.500e-01]	
Maximum	[6.197e-17 5.000e-01 0.000e+00]		Maximum	[6.197e-17 5.000e-01 0.000e+00]		Maximum	[6.197e-17 5.000e-01 0.000e+00]	
Participant: MAPDL-2			Participant: MAPDL-2			Participant: MAPDL-2		
Number of cell regions	1		Number of cell regions	1		Number of cell regions	1	
Number of cells	2 088		Number of cells	2 088		Number of cells	2 088	
Hexahedral20	2 088		Hexahedral20	2 088		Hexahedral20	2 088	
Number of faces	6 666		With warped faces	456		With warped faces	2 088	
Volume (m3)	9.790e-02		Volume (m3)	6 666		Volume (m3)	6 666	
Bounding Box (m)			Bounding Box (m)			Bounding Box (m)		
Minimum	[-2.500e-01 0.000e+00 -2.500e-01]		Minimum	[-2.500e-01 0.000e+00 -2.500e-01]		Minimum	[-2.500e-01 0.000e+00 -2.500e-01]	
Maximum	[2.500e-01 5.000e-01 2.500e-01]		Maximum	[2.500e-01 5.000e-01 2.500e-01]		Maximum	[2.500e-01 5.000e-01 2.500e-01]	
Total			Total			Total		
Number of cells	3 080		Number of cells	3 080		Number of cells	3 080	
Number of faces	9 944		Number of faces	9 944		Number of faces	9 944	
Number of nodes	10 947		Number of nodes	10 947		Number of nodes	10 947	
Instancing and/or Reference Frame transformations are defined for this analysis. Note that the mesh statistics report non-instanced/untransformed geometries.			Instancing and/or Reference Frame transformations are defined for this analysis. Note that the mesh statistics report non-instanced/untransformed geometries.			Instancing and/or Reference Frame transformations are defined for this analysis. Note that the mesh statistics report non-instanced/untransformed geometries.		
MAPPING SUMMARY			MAPPING SUMMARY			MAPPING SUMMARY		
	Source	Target		Source	Target		Source	Target
CouplingInterface 1			CouplingInterface 1			CouplingInterface 1		
DataTransfer 1			DataTransfer 1			DataTransfer 1		
Mapped Volume [%]	100	100	Mapped Volume [%]	100	100	Mapped Volume [%]	100	100
Mapped Elements [%]	100	100	Mapped Elements [%]	100	100	Mapped Elements [%]	100	100
Mapped Nodes [%]	100	94	Mapped Nodes [%]	100	94	Mapped Nodes [%]	100	100

Log file mesh statics and mapping summary comparison

Mapping Accuracy Improvements

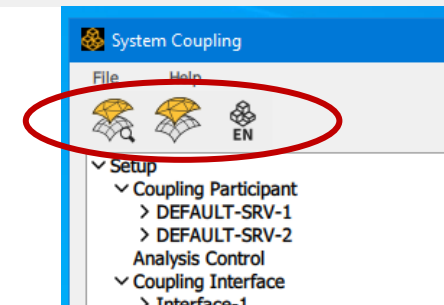
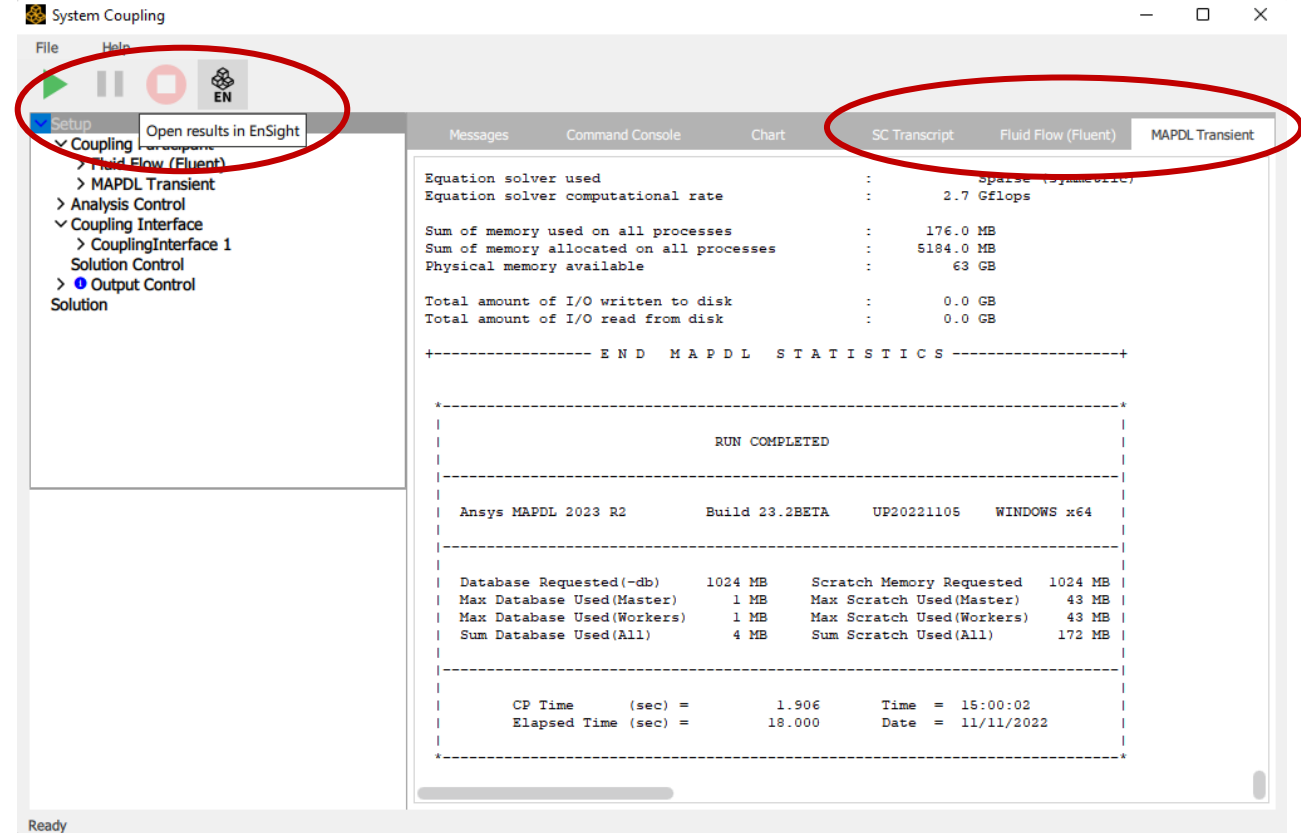
- Implemented new “Source Element RBF” algorithm
 - Directly uses source data if it’s on elements
- Fully eliminated node-element reconstruction steps for profile-preserving volume mapping algorithms



Target without reconstruction (note how the data is smeared)

User Interface Improvements

- Participant solver transcripts available in System Coupling GUI
 - Provides a more comprehensive view of the coupled analysis inside one GUI
 - System Coupling transcript (.scl file) is also available
 - Participant APIs updated to allow participants to provide transcript file name
- GUI toolbar buttons
 - Common operations available with one click (Solve, Open Results in EnSight, etc.)
 - More to come in future releases...



User Interface Improvements

- Custom participant partitioning for distributed parallel
 - Allows full flexibility for distributing participants for parallel runs
 - Call **GetMachines()** query to get the list of machines available at run-time
 - Useful if machines unknown ahead of time, e.g. HPC cluster environment with job scheduler
 - Use Python script to assign cores and machines to participants
 - Call **PartitionParticipants()** command takes partitioning information from above and translates it to participant execution commands

Example script:

```
allMachines = GetMachines()
m1 = allMachines[0]
m2 = allMachines[1]
m3 = allMachines[2]

aedtMachines = []
aedtMachines.append(m1.copy())
aedtMachines.append(m2.copy())

fluentMachines = []
fluentMachines.append(m1.copy())
fluentMachines.append(m2.copy())
fluentMachines.append(m3.copy())

partitioningInfo = {}
partitioningInfo['AEDT-1'] = aedtMachines
partitioningInfo['FLUENT-2'] = fluentMachines

PartitionParticipants(PartitioningInfo = partitioningInfo)
```

Execution Information

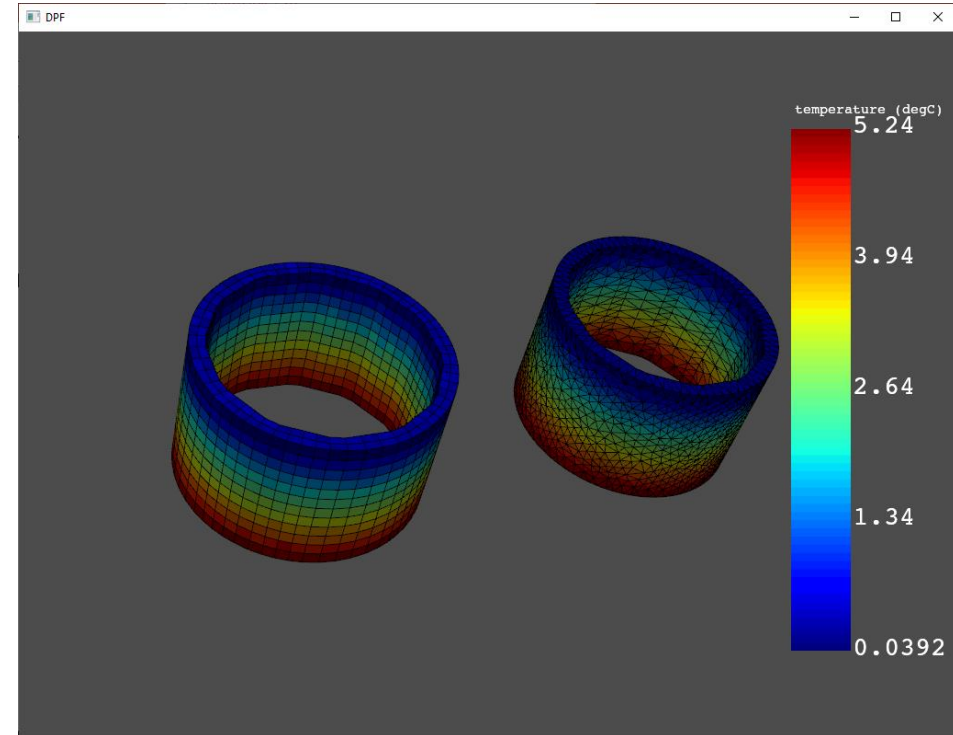
```
System Coupling
Command Line Arguments:
--cnf=lebvmsles124ugel:8,lebvmsles124uge2:8,lebvmsles124uge3:8 -R run.py
Working Directory:
/net/nfs.lebisilon/home/aalhoura/tests/custompartitioning/aedt-2d-transient-fluent-3d-steady-auto

Ansys Electronics Desktop
Execution Command:
"/net/nfs.lebisilon/home/aalhoura/AnsysEM/v231/Linux64/ansysedt" -ng -f
eatures=SF6694_NON_GRAPHICAL_COMMAND_EXECUTION -distributed -auto -mach
inelist list=lebvmsles124ugel:-1:8,lebvmsles124uge2:-1:8 -sctest 44391
-schost lebvmsles124ugel.ansys.com -scname "AEDT-1" --runscript "Electri
cMotor_2D_Transient_SystemCouplingSetup1.py"
Working Directory:
/net/nfs.lebisilon/home/aalhoura/tests/custompartitioning/aedt-2d-transient-fluent-3d-steady-auto/Maxwell

Fluid Flow (Fluent)
Execution Command:
"/net/nfs.lebisilon/home/aalhoura/ansys_inc/v231/fluent/bin/fluent" 3dd
p -g -sctest=44391 -schost=lebvmsles124ugel.ansys.com -scname="FLUENT-2
" -i FLUENT-2.jou -t24 --cnf=lebvmsles124ugel:8,lebvmsles124uge2:8,lebv
msles124uge3:8
Working Directory:
/net/nfs.lebisilon/home/aalhoura/tests/custompartitioning/aedt-2d-transient-fluent-3d-steady-auto/Fluent
```

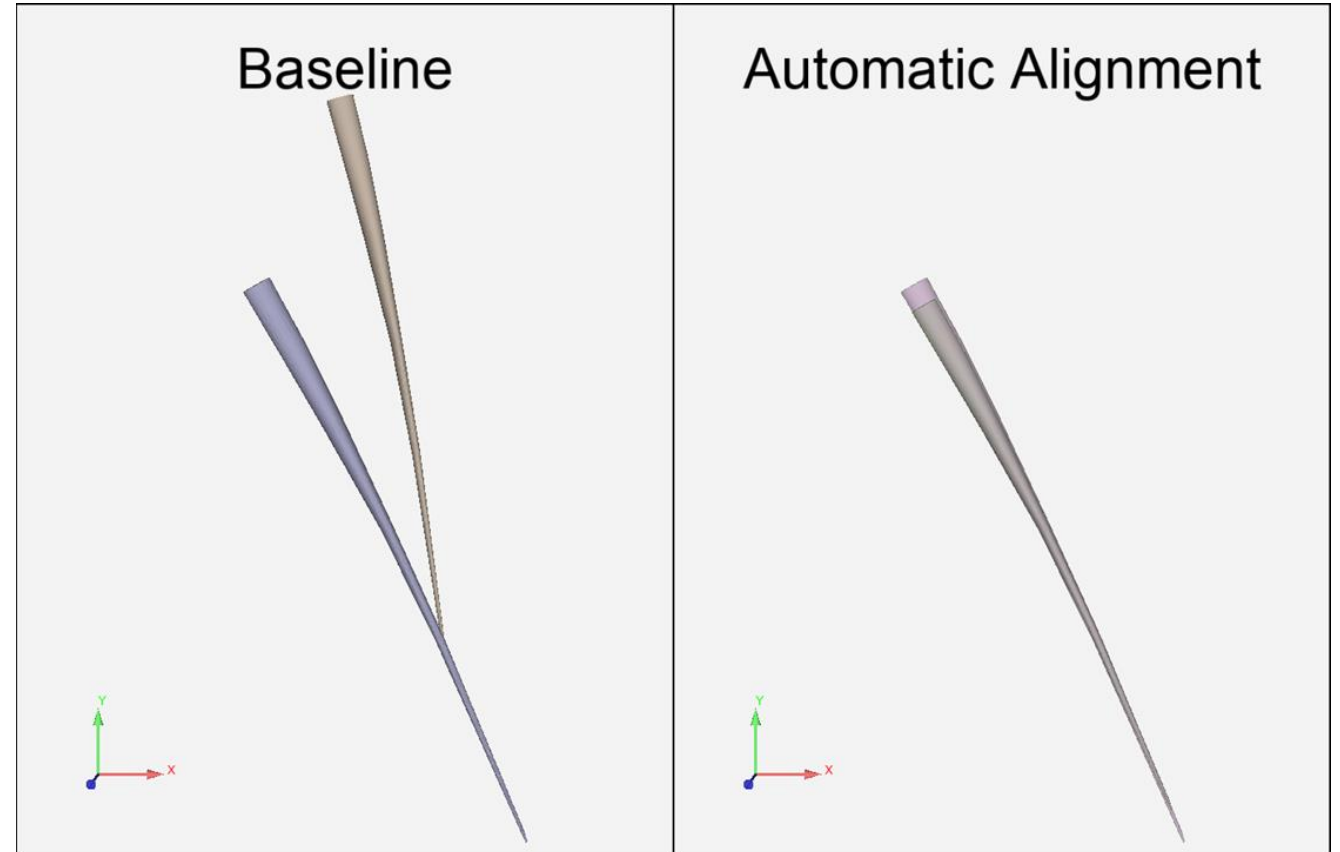
DPF Mapping Operators

- System Coupling mapping library embedded inside DPF operators
- New DPF operators
 1. `sc_mapping`
 - Simplified workflow version to map one variable easily
 2. `create_sc_mapping_workflow + mapping_workflow`
 - Separates mapping weights calculation and interpolation
 - Allows interpolating multiple variables efficiently
- Available mapping features
 - Surface, volume
 - Polyhedral meshes
 - Meshes with high order elements
 - Point cloud mapping
 - Vectors, scalars
 - Conservative, profile-preserving mapping



Automatic Interface Alignment (Beta)

- Align source and target geometry automatically
- Rigid body alignment (translation and rotation)
- Can work together with manual transformations, e.g.
 1. Manual transformation
 2. Automatic alignment
 3. Manual transformation
- Work in progress – more to come in 2023 R2



 **Ansys**

