The ICoCo based Coupling of TRACE and SCF in SALOME, and a new function of TRACE: an automatic connection between Cylinder and Cartesian VESSELs instead of using the VESSEL JUNCTION component in SNAP

Kanglong Zhang
Technical Advisor: Dr. Victor Hugo Sanchez Espinoza

Institute for Neutron Physics and Reactor Technology (INR)

Presentation Outline

Karlsruhe Institute of Technology

The Coupling of TRACE and SCF through ICoCo in SALOME

- Elements involved in the coupling system
- Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME
- Strategy of the Coupling System
- Verification of the Coupling System

Presentation Outline

The Coupling of TRACE and SCF through ICoCo in SALOME

- ***Elements involved in the coupling system***
- **Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME**
- **Strategy of the Coupling System**
- **Verification of the Coupling System**

Elements involved in the coupling system

Karlsruhe Institute of Technology

Title: The Coupling of **TRACE** and **SCF** through **ICoCo** and **SALOME**

TRACE: the thermal-hydraulic system code by U.S. NRC.

SCF: the sub-channel code by KIT Germany.

ICoCo: a code coupling framework initiated by CEA France.

SALOME: an open-source platform for numerical simulations by CEA and EDF.

ICoCo -- two prerequisites

1) Codes must have explicit meshes in MED format.
2) Codes must be pre-split into specified functional components.

- **setDataFile** → Specify the input, output, and arguments.
- **initialize** → Run initialization and construct the mesh.
- **computeTimeStep** → Calculate the timestep size.
- **initTimeStep** → Initialize the timestep size.
- **solveTimeStep** → Run the current step.
- **getOutputMEDField** → Extract the desired field in MED format.
- **getInputMEDFieldTenplate** → Get the template receiving the data from another code.
- **setInputMEDField** → Write the MED field from another code to memory.
- **terminate** → Terminate the calculation.

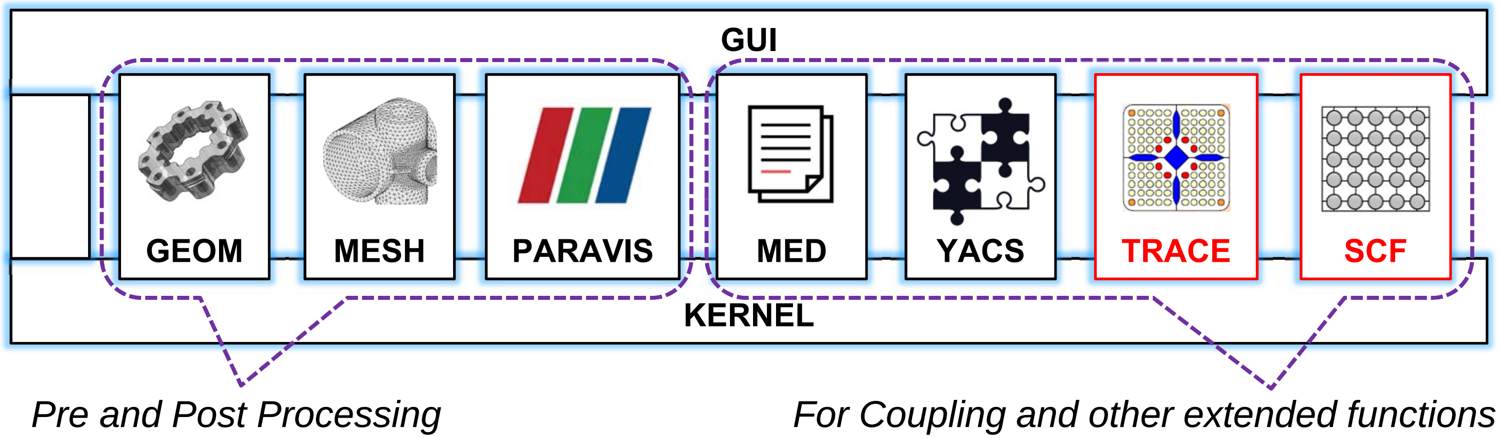
Elements involved in the coupling system

Title: The Coupling of **TRACE** and **SCF** through **ICoCo** and **SALOME**

TRACE: the thermal-hydraulic system code by U.S. NRC.
SCF: the sub-channel code by KIT Germany.
ICoCo: a code coupling framework initiated by CEA France.
SALOME: an open-source platform for numerical simulations by CEA and EDF.

SALOME -- two main functions

1) Provide generic functions for numerical pre- and post- processing.
2) Manage codes coupling, corresponds to ICoCo.



TRACE and SCF should be implemented into SALOME based on ICoCo standard.

Presentation Outline

The Coupling of TRACE and SCF through ICoCo in SALOME

- Elements involved in the coupling system
- *Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME*
- Strategy of the Coupling System
- Verification of the Coupling System

Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME

Karlsruhe Institute of Technology

sou1.f90
sou2.f90
sou3.f90
…

Code original source

Re-written
Re-organize

sour1_salome.f90
sour2_salome.f90
sour3_salome.f90
…

CODE-SALOME source

+

interface.hxx
interface.cxx
…

C++ envelope or wrapper

CODE-SALOME Internal Object

+

SWIG_inter.i

SWIG file

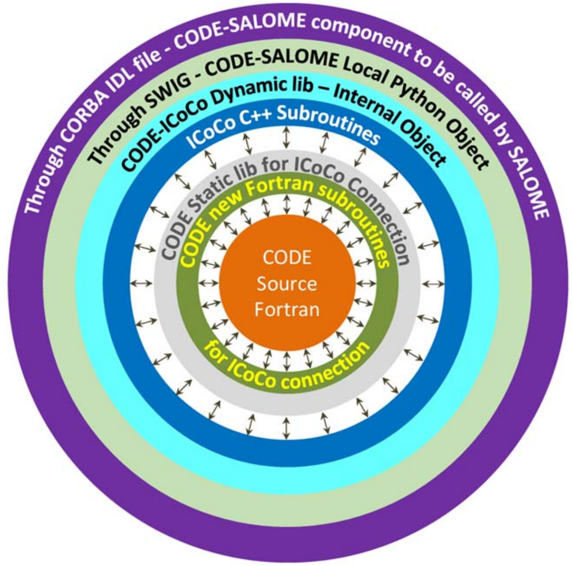
CODE-SALOME Local Python Object

+

CORBA.idl

CORBA IDL file

CODE-SALOME Component



Four Steps to implement TRACE to SALOME
And
Four Levels of TRACE Objects

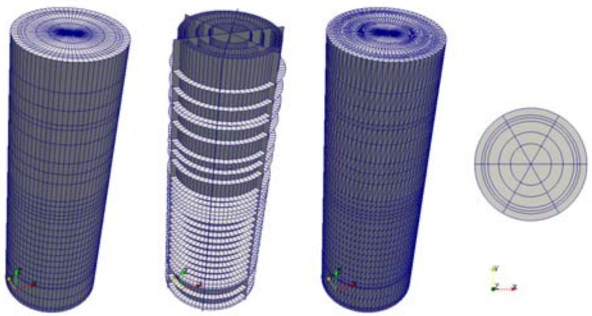
1) *Based on code original source, re-write and re-organize the files to develop the CODE-SALOME source.*
2) *Develop the functional interface files which are directly used in YACS. Develop the MED-related functions.*
3) *Write the file which is used by SWIG to wrap the C++ files to Python. Then, make the local Python object of TRACE in SALOME.*
4) *Wrap the Python Object and to the CORBA remote object of in SALOME.*

Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME

Karlsruhe Institute of Technology

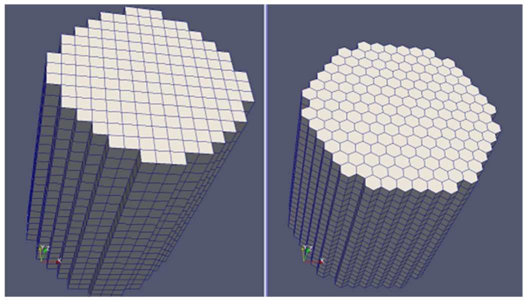
1) The TRACE and SCF modules in SALOME fully follows the ICoCo framework and contain the functional components.
2) Their different meshes have to map to translate and transfer fields.
3) <u>A toolkit for mesh interpolation and field mapping is desired.</u>

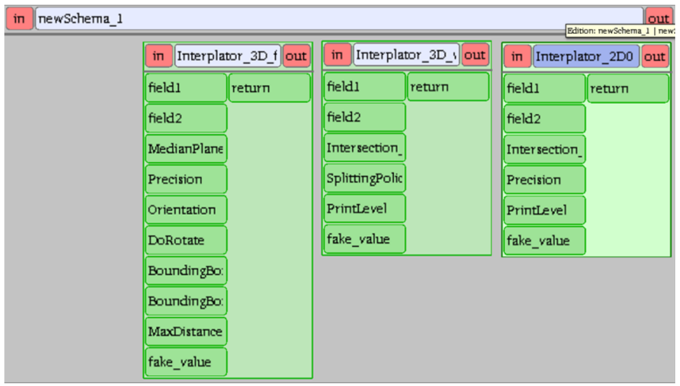
TRACE MED meshes



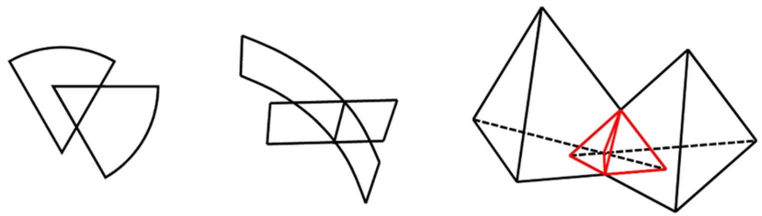
Mapping

SCF MED meshes



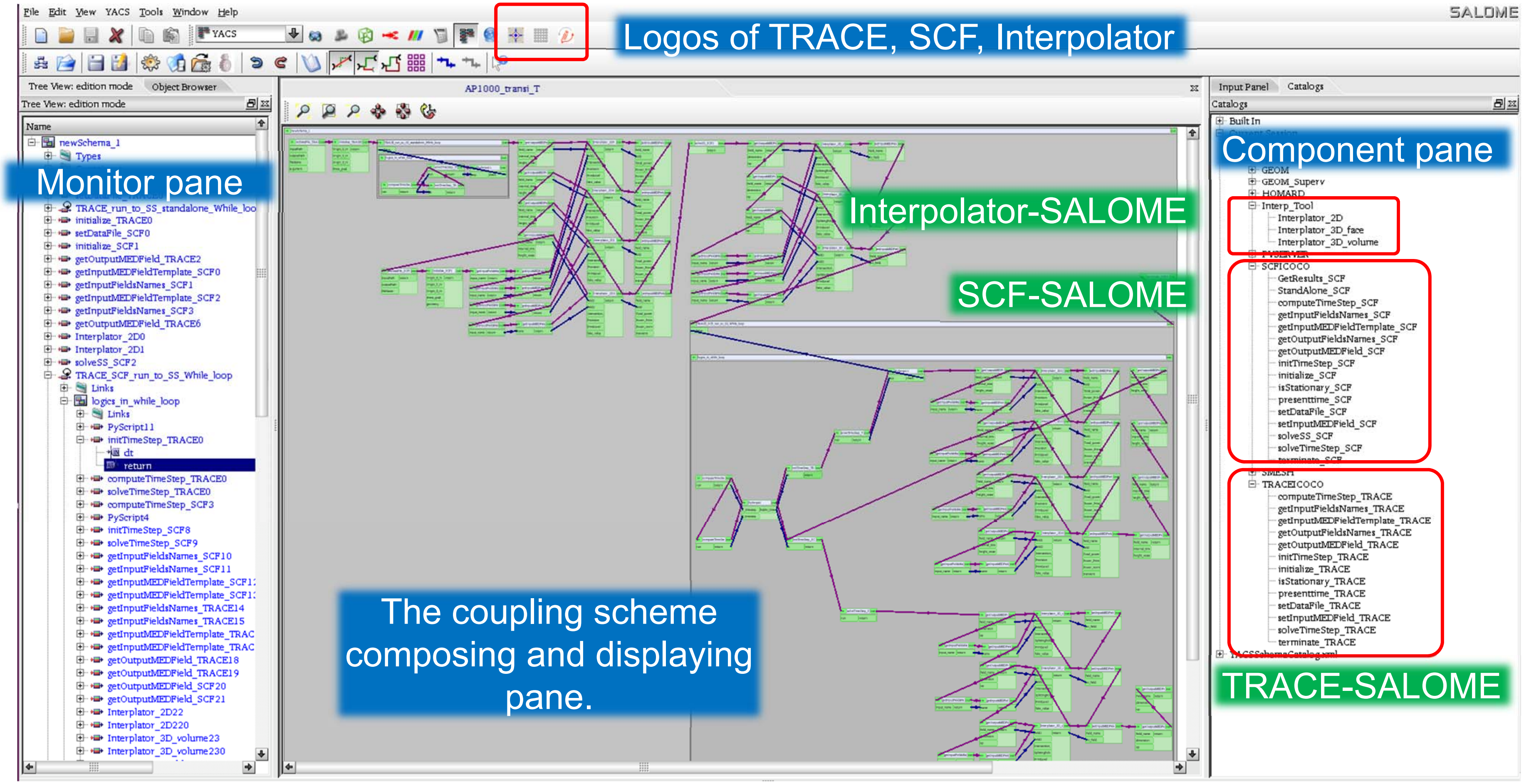


Interpolation toolkit



2D Interp 3D face Interp 3D volume Interp

Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME

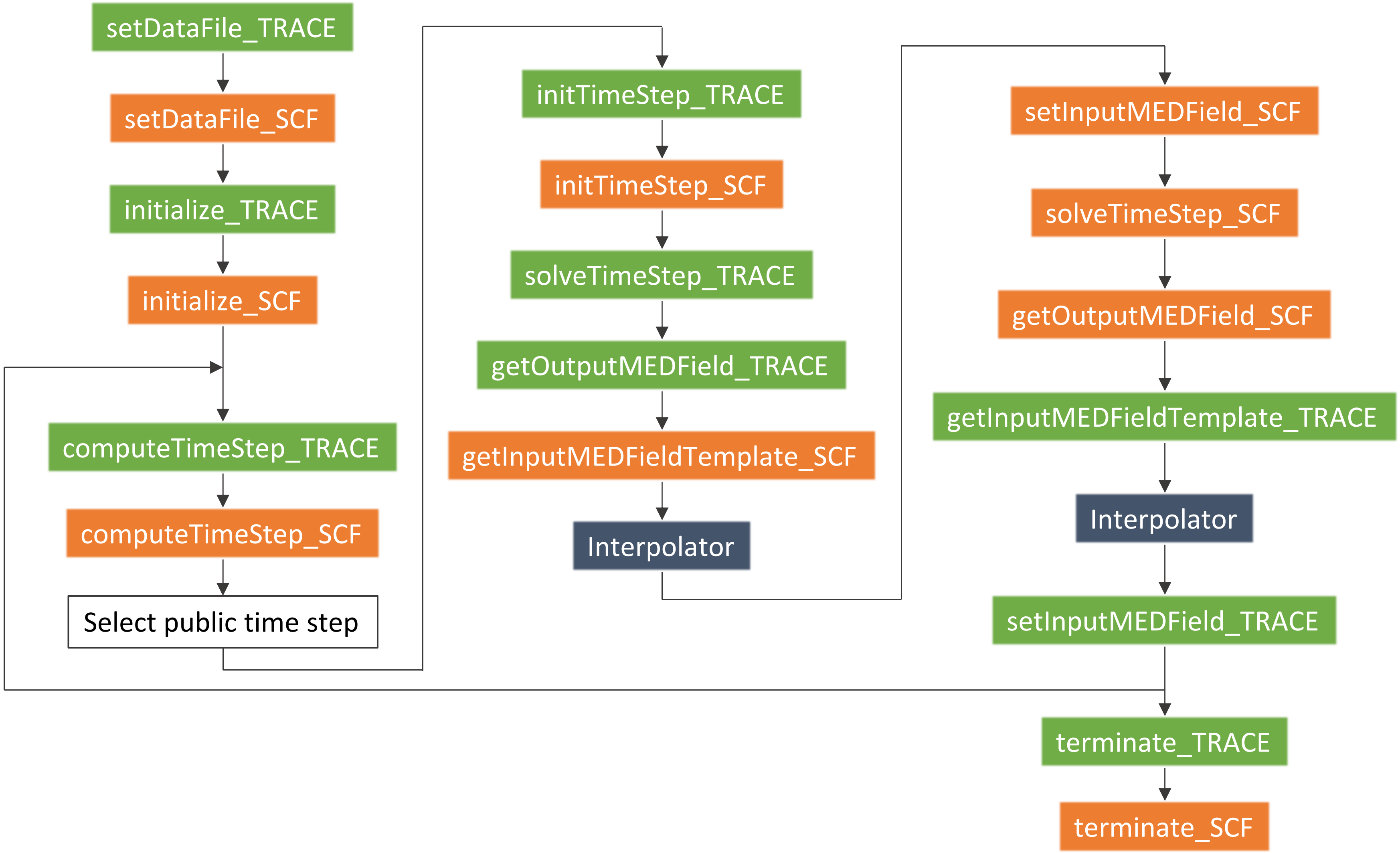


Presentation Outline

The Coupling of TRACE and SCF through ICoCo in SALOME

- **Elements involved in the coupling system**
- **Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME**
- ***Strategy of the Coupling System***
- **Verification of the Coupling System**

Strategy of the Coupling System



Presentation Outline

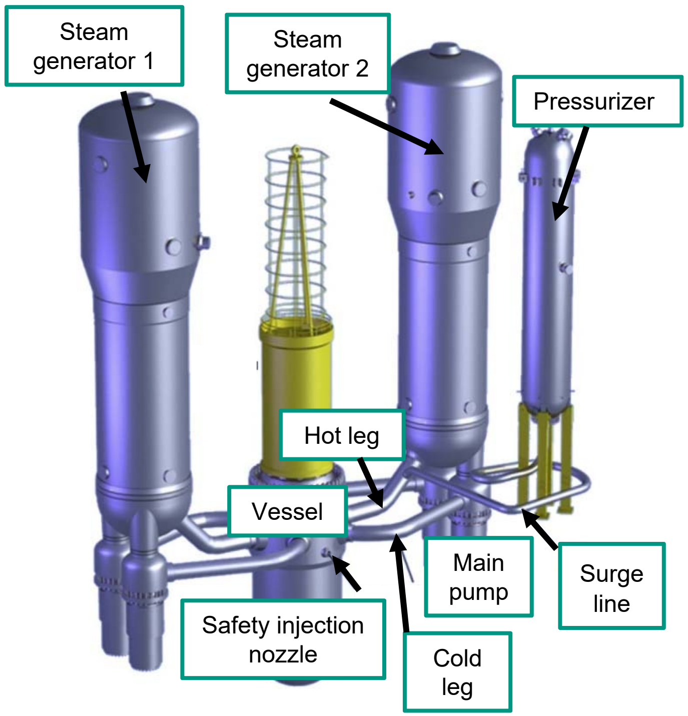
Karlsruhe Institute of Technology

The Coupling of TRACE and SCF through ICoCo in SALOME

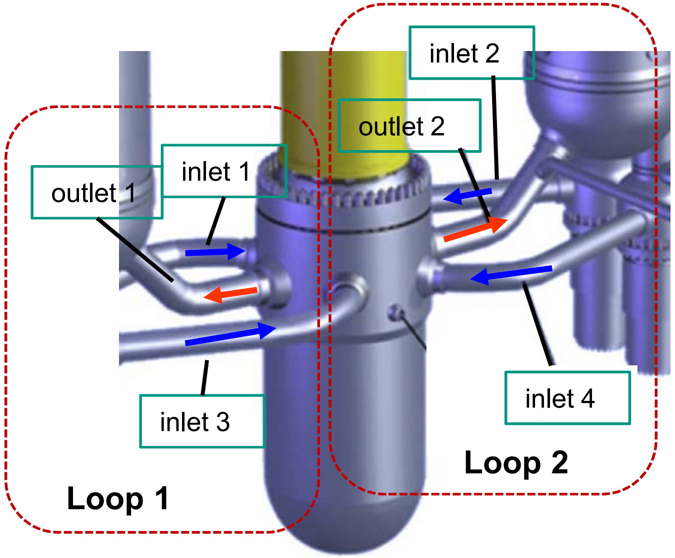
- Elements involved in the coupling system
- Implementation of TRACE-ICoCo, SCF-ICoCo, and an interpolation toolkit to SALOME
- Strategy of the Coupling System
- ***Verification of the Coupling System***

Verification of the Coupling System

An AP1000 transient case:



Configuration of the main components of AP1000 reactor



Main initial parameters

1) Core power: 3400 MW
2) Core mass flow rate: 14275 kg/s
3) Core inlet temperature: 553.8 K
4) Core outlet temperature: 596.5 K

Transient sequence

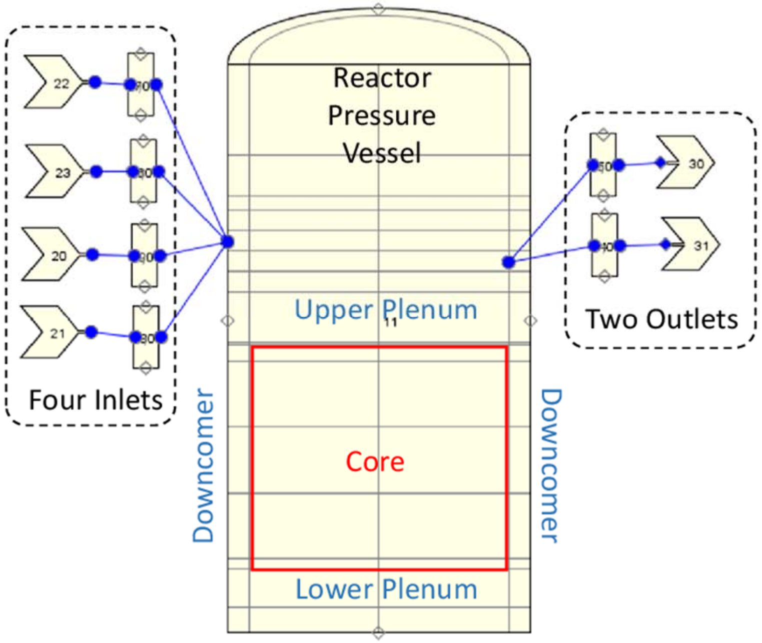
1) t = 0 s, temp (inlet 1, 2, 3, 4) = 553.7 K
2) t = 5 s, temp (inlet 1, 2, 3, 4) = 553.7 K
3) t = 15 s, temp (inlet 1, 3) = 503.7 K
4) t = 20 s, temp (inlet 1, 3) = 503.7 K

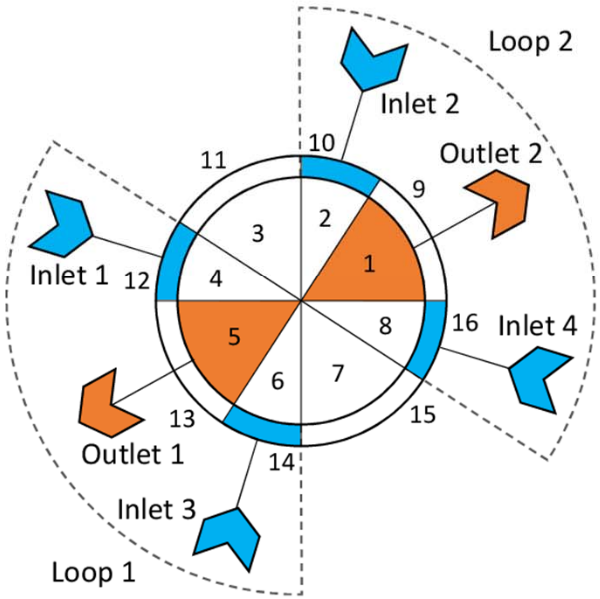
Verification of the Coupling System

TRACE model

The model was develop with SNAP for TRACE v5.1051

Axial levels	16
Radial rings	2
Azimuthal sectors	8





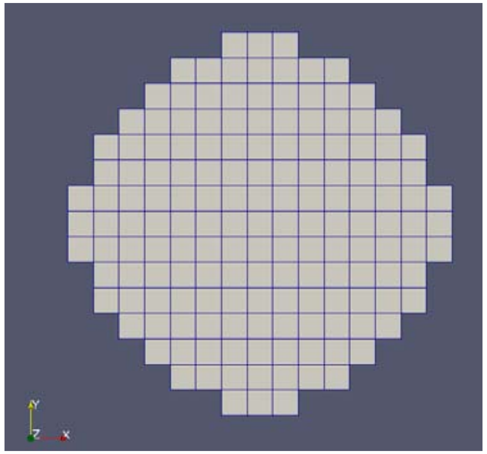
Verification of the Coupling System

Karlsruhe Institute of Technology

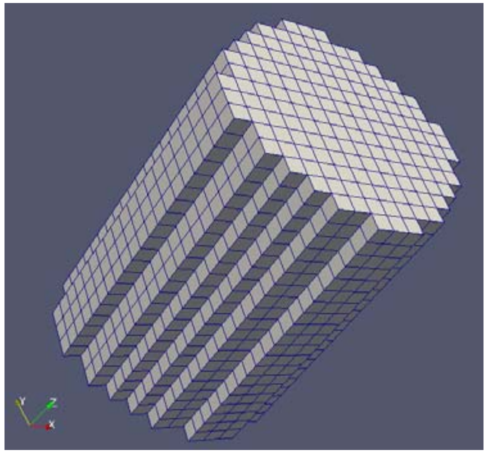
SCF model

The model was develop with SCF v3.3

total_fuel_assemblies	157
fuel_assembly_pitch $[m]$	0.214
rod_diameter $[m]$	0.0095
channel_area $[m^2]$	0.0253127
wetted_perimeter $[m]$	9.4808
heated_perimeter $[m]$	7.878782
channel_gap $[m]$	1.016×10^{-3}
total_axial_length $[m]$	4.2671



2D mesh



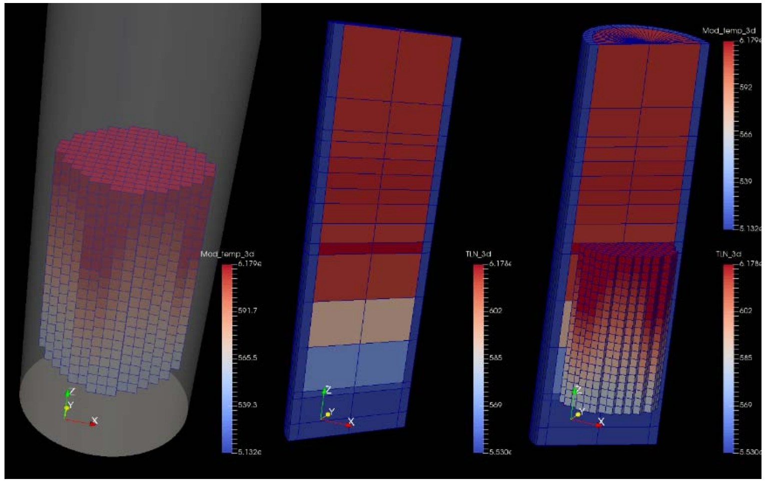
3D mesh

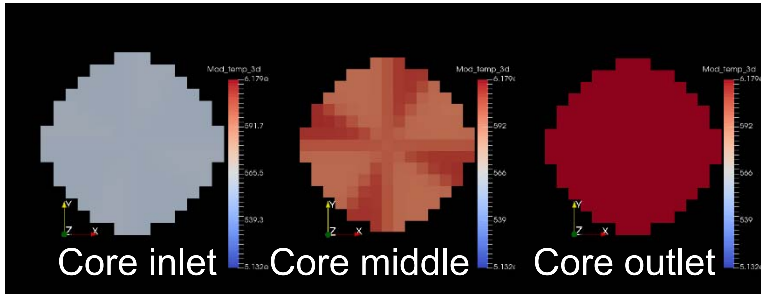
Verification of the Coupling System

Karlsruhe Institute of Technology

The coolant temperature profile at the beginning of the transient

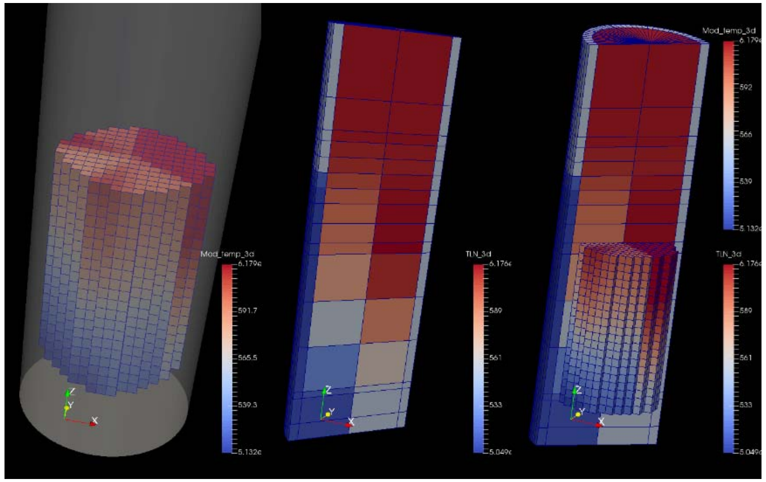
The coolant temperature distribute symmetrically in TRACE(Vessel) and SCF(Core).

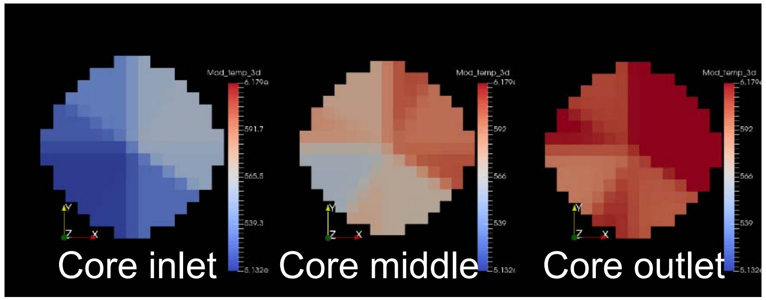




The coolant temperature profile at the end of the transient

The coolant temperature at the loop1 corresponding area of TRACE(Vessel) and SCF(Core) is lower than loop2 area.

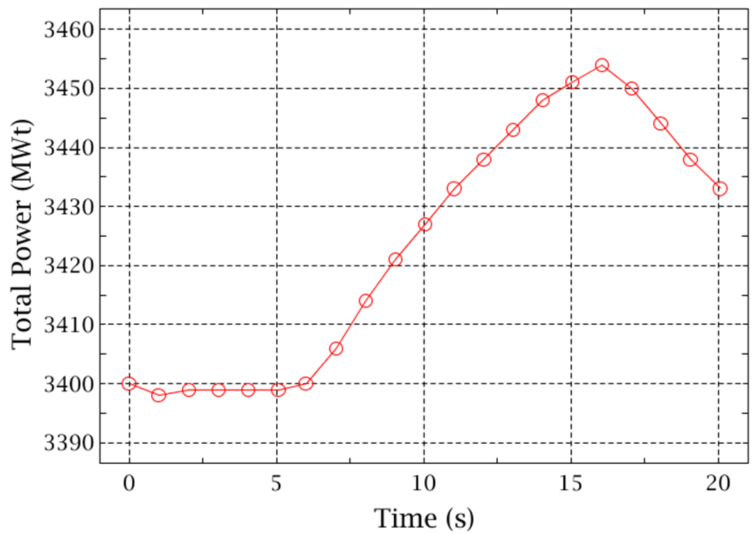




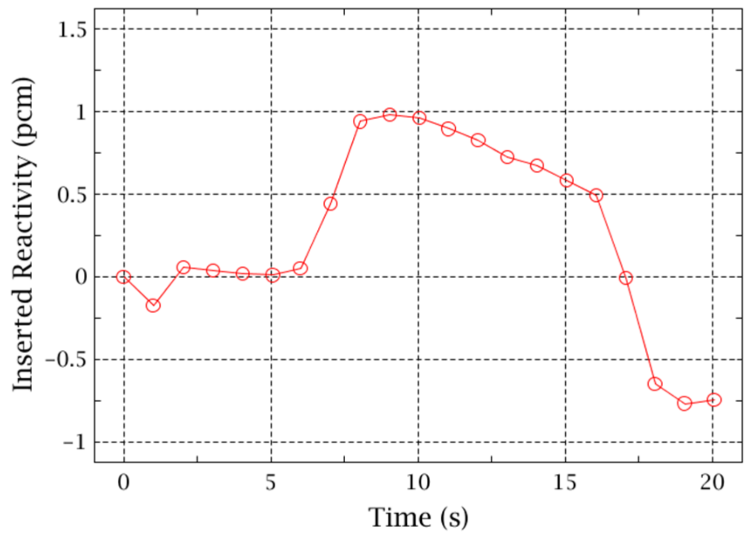
Verification of the Coupling System

Karlsruhe Institute of Technology

The neutronic results of the transient by SCF (point kinetic model)



Evolution of the total thermal power in the core



Evolution of the reactivity in the core

1) As the coolant temperature decreases, a positive reactivity was introduced to the core. It first increase rapidly and then gradually reduce and finally become negative because of the thermal feedback.
2) Corresponding to the evolution of reactivity, the thermal power of the core increases to the peak and then decreases.

Presentation Outline

Karlsruhe Institute of Technology

A new function of TRACE: an automatic connection between Cylinder and Cartesian VESSELs instead of using the VESSEL JUNCTION component in SNAP

- Current coupling method between Cylinder and Cartesian vessels
- The new automatic coupling between the vessels
- Additional features for the vessel coupling cases

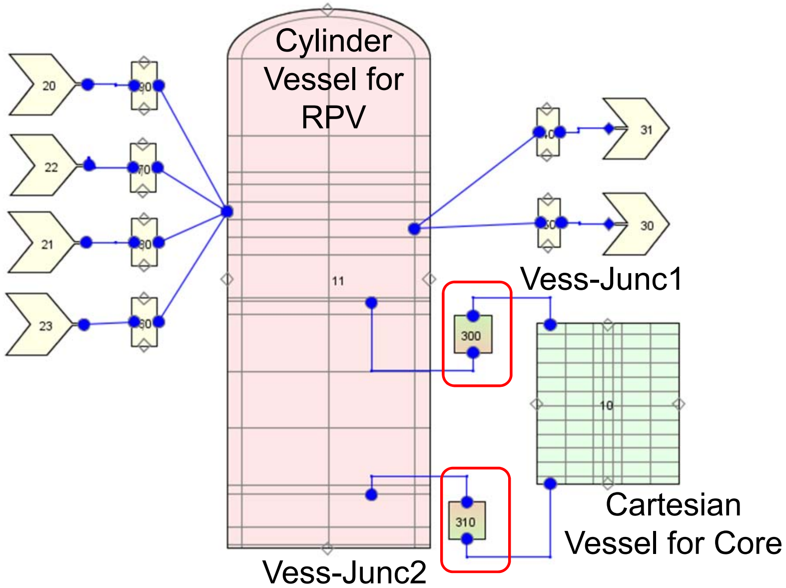
Presentation Outline

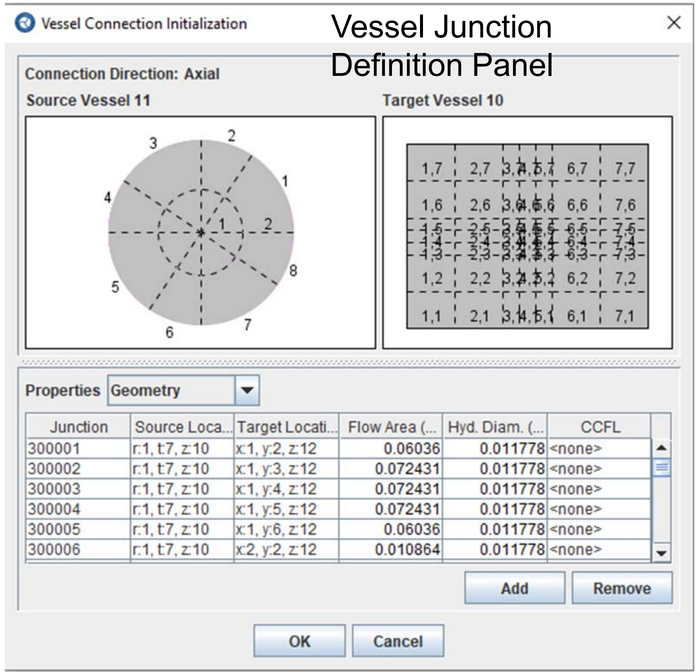
A new function of TRACE: an automatic connection between Cylinder and Cartesian VESSELs instead of using the VESSEL JUNCTION component in SNAP

- ***Current coupling method between Cylinder and Cartesian vessels***
- **The new automatic coupling between the vessels**
- **Additional features for the vessel coupling cases**

Current coupling method between Cylinder and Cartesian vessels

A TRACE model includes Cylinder and Cartesian vessels





1) The Cylinder Vessel models the RPV.
2) The Cartesian Vessel models the Core.
3) The two vessel are connected by two vessel junction component at core inlet and outlet.

1) Users define the positional correspondence.
2) Users calculate and set the flow area, hydraulic diameters for each channel.
3) Both of the operations are boring and error-prone.

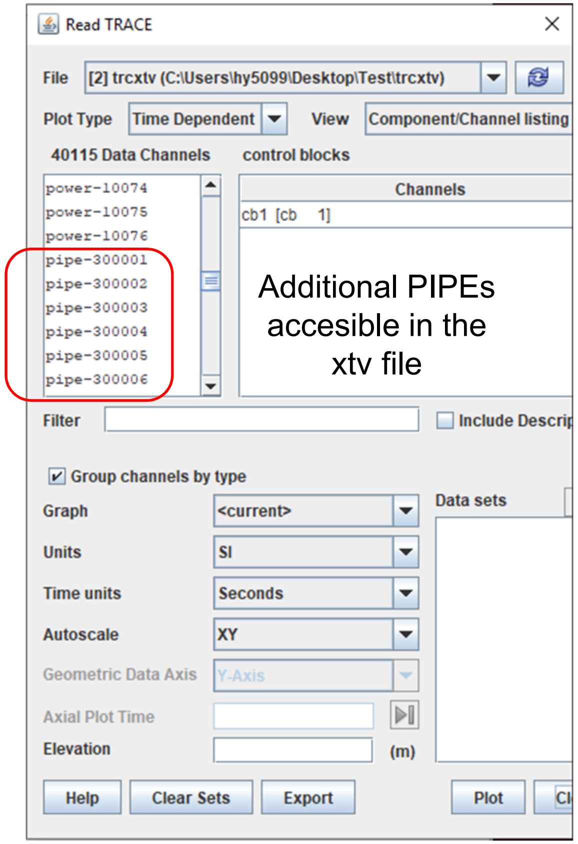
Current coupling method between Cylinder and Cartesian vessels

Input file of the vessel coupling case

```
*************************
* component-number data *
*************************
*
*  Component input order (IORDER)
*-- type ---- num ------------- name ---------------- +    jun1   jun2   jun3
* VESSEL   *      10 s * core                         +
* VESSEL   *      11 s *                              +
* FILL     *      20 s *                              +     110
* FILL     *      21 s *                              +     100
* FILL     *      22 s *                              +      90
* FILL     *      23 s *                              +      10
* BREAK    *      30 s *                              +      20
* BREAK    *      31 s *                              +      80
* PIPE     *      40 s *                              +      70     80
* PIPE     *      50 s *                              +      30     20
* PIPE     *      60 s *                              +      10    120
* PIPE     *      70 s *                              +      90    130
* PIPE     *      80 s *                              +     100    140
* PIPE     *      90 s *                              +     110    150
* PIPE     * 300001 s * $300$ vessel 11 to vessel 10  + 300001130000l2
* PIPE     * 300002 s * $300$ vessel 11 to vessel 10  + 30000213000022
* PIPE     * 300003 s * $300$ vessel 11 to vessel 10  + 30000313000032
* PIPE     * 300004 s * $300$ vessel 11 to vessel 10  + 30000413000042
* PIPE     * 300005 s * $300$ vessel 11 to vessel 10  + 30000513000052
* PIPE     * 300006 s * $300$ vessel 11 to vessel 10  + 30000613000062
* PIPE     * 300007 s * $300$ vessel 11 to vessel 10  + 30000713000072
```

Additonal PIPEs by vessel junction component

1) A series of additional PIPE components are added to the input file. They appear in the component list and more lines follow in the component definition data block.
2) Make the input file redundant and unreadable.



1) Uses can access the variables of each additional PIPE by the vessel junction.

Presentation Outline

A new function of TRACE: an automatic connection between Cylinder and Cartesian VESSELs instead of using the VESSEL JUNCTION component in SNAP

- **Current coupling method between Cylinder and Cartesian vessels**
- ***The new automatic coupling between the vessels***
- **Additional features for the vessel coupling cases**

The new automatic coupling between the vessels

Karlsruhe Institute of Technology

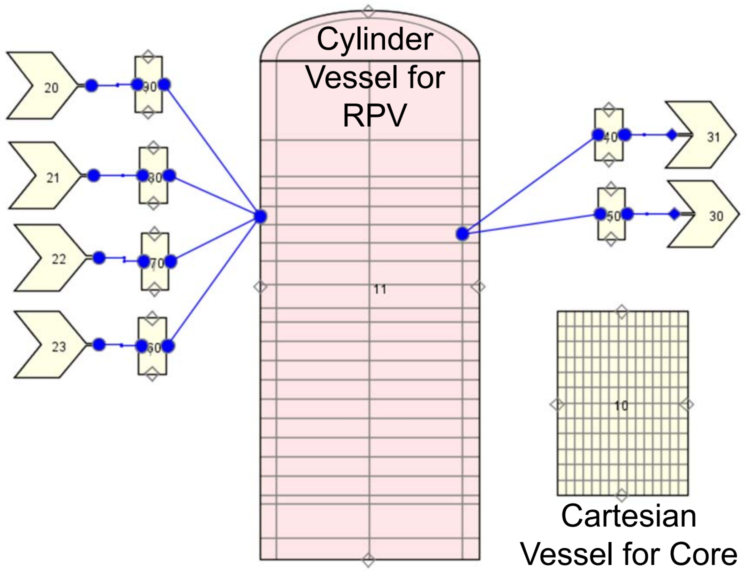
Drawbacks of current vessel coupling method

1) Manual definition of posi corre. → 1) Automatic definition of posi corre.
2) Manual calculation of FA, HD. → 2) Automatic calculation of FA, HD.
3) Redundant input file. → 3) Clean input file.

Features of the new vessel coupling method

1) Automatic definition of posi corre.
2) Automatic calculation of FA, HD.
3) Clean input file.

A TRACE model includes Cylinder and Cartesian vessels



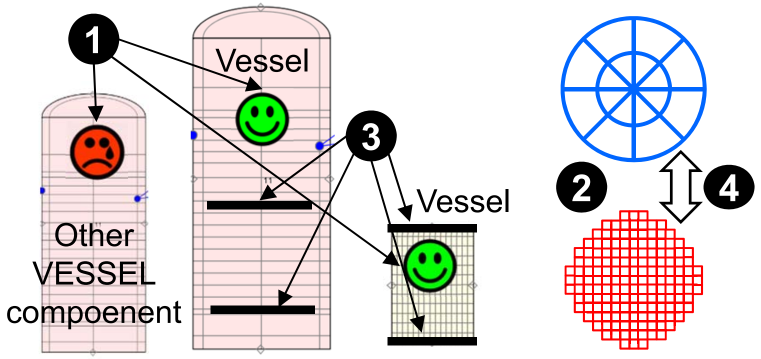
Modeling instruction for users

1) Both of the Cylinder and Cartesian vessels must include core boundary definition.
2) The axial flow area of the core area in the Cylinder vessel must be 0.0.
3) The largest length (x, y direction) of Cartesian vessel must not exceed the core outer radial ring of the Cylinder vessel.
4) Source the environment of MED libs.
5) Restart calculation is available.

The new automatic coupling between the vessels

Principles of the method

1) Recognize the VESSEL component which models the Vessels.
2) Read the geometry data of the Vessels and construct the 2D mesh based on MED libs.
3) Locate the core inlet and outlets.
4) Load the flow area data to meshes and map the 2D meshes.
5) Build new PIPE as junctions between vessels and insert to TRACE data structure.



Source code level (only arrays)

1) vessTab%igeom, vessTab%vessType.
2) vessTab%nasx, %nrsx, %ntsx, vsAr%z(:), %rad(:), %th(:).
3) vessTab%icru, %icrl.
4) vaAr3%faz(:, :, :).
5) vessTab%ncsr, vsAr%isrl(:), vsAr%isrc(:), vsAr%isrf(:), vsAr%juns(:).
6) Other arrays to be modified in TRACE data structure: compIndices(:), compNumList(:), compCcoList(:), g1DAr(:), intAr(:), PipeAr(:), genTab(:), PipeTab(:), fillTab(:), fillAr(:), breakTab(:), breakAr(:), chanTab(:), chanAr(:), radTab(:), radar(:), pumpTab(:), pumpAr(:), valveTab(:), valveAr(:), plenTab(:), plenAr(:), turbTab(:), prizeTab(:), teeTab(:), teeAr(:), heatrTab(:), heatrAr(:), jetpTab(:), sepdAr(:), hsTab(:), hsAr(:), hsTab(:), hsAr(:), powTab(:), powAr(:), flPowTab(:), flPowAr(:), exterTab(:), exterAr(:), vsAr2(:), vsAr4(:), vsSrcAr(:).

The new automatic coupling between the vessels

Karlsruhe Institute of Technology

- trac → input
  - → CompDat — med_TracInputM.f90
    - → ReSetVariables — med_func.f90, med_data.f90
    - → RVssl — med_VessInputM.f90
    - → rdcomp
      - → rpipe
        - → pipeHeadr — med_PipeInputM.f90
        - → rcomp — med_rcomp.f90
  - → RdTprRest — med_TprM.f90
    - → RdTprVessel — med_TprVesselM.f90

Logical procedures of the automatic coupling functions in the

Original TRACE subs & files

Modified TRACE subs & files

Newsubs & files

1) 13 modified TRACE source files
2) 2 new files

med_PipeArrayM.f90

med_CountingCompM.f90

med_IntArrayM.f90

med_Global.f90

med_FltM.f90

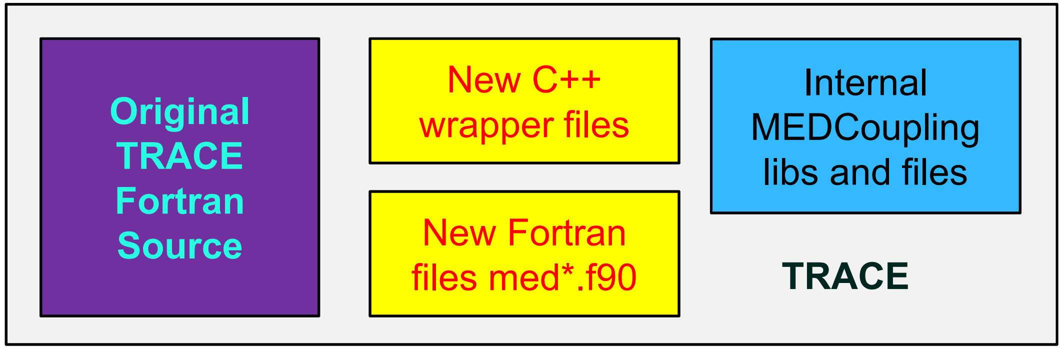
med_Gen1DArrayM.f90

med_PipeVltM.f90

The new automatic coupling between the vessels

Python scons/scons.py compiler = intel icoco_med = med

In addition to TRACE normal functions, it can automatically couple the Cylinder and Cartesian vessels.



Python scons/scons.py compiler = intel

Compile Normal TRACE executable.

Presentation Outline

A new function of TRACE: an automatic connection between Cylinder and Cartesian VESSELs instead of using the VESSEL JUNCTION component in SNAP

- **Current coupling method between Cylinder and Cartesian vessels**
- **The new automatic coupling between the vessels**
- ***Additional features for the vessel coupling cases***

Additional features for the vessel coupling cases

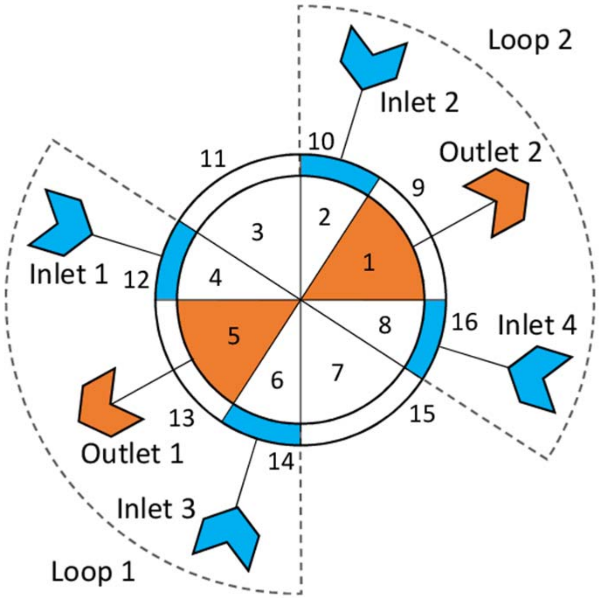
Karlsruhe Institute of Technology

1) Full automation of simulation for vessel involved cases.

VESSEL for RPV? → Yes → Only one Cylinder RPV → No RPV coupling; Post-process Cylinder RPV.

VESSEL for RPV? → Yes → Cylinder&Cartesian RPVs → Automatic RPV coupling; Post-process Cylinder and Cartesian RPVs.

VESSEL for RPV? → No → Run as normal TRACE.



2) Post-processing for the Cylinder Vessel and Cartesian Vessel

Testing Case – AP1000 reactor

2.1 Steady State calculation

Temperature (inlet 1, 2, 3, 4) = 553.7 K

2.2 Restart calculation

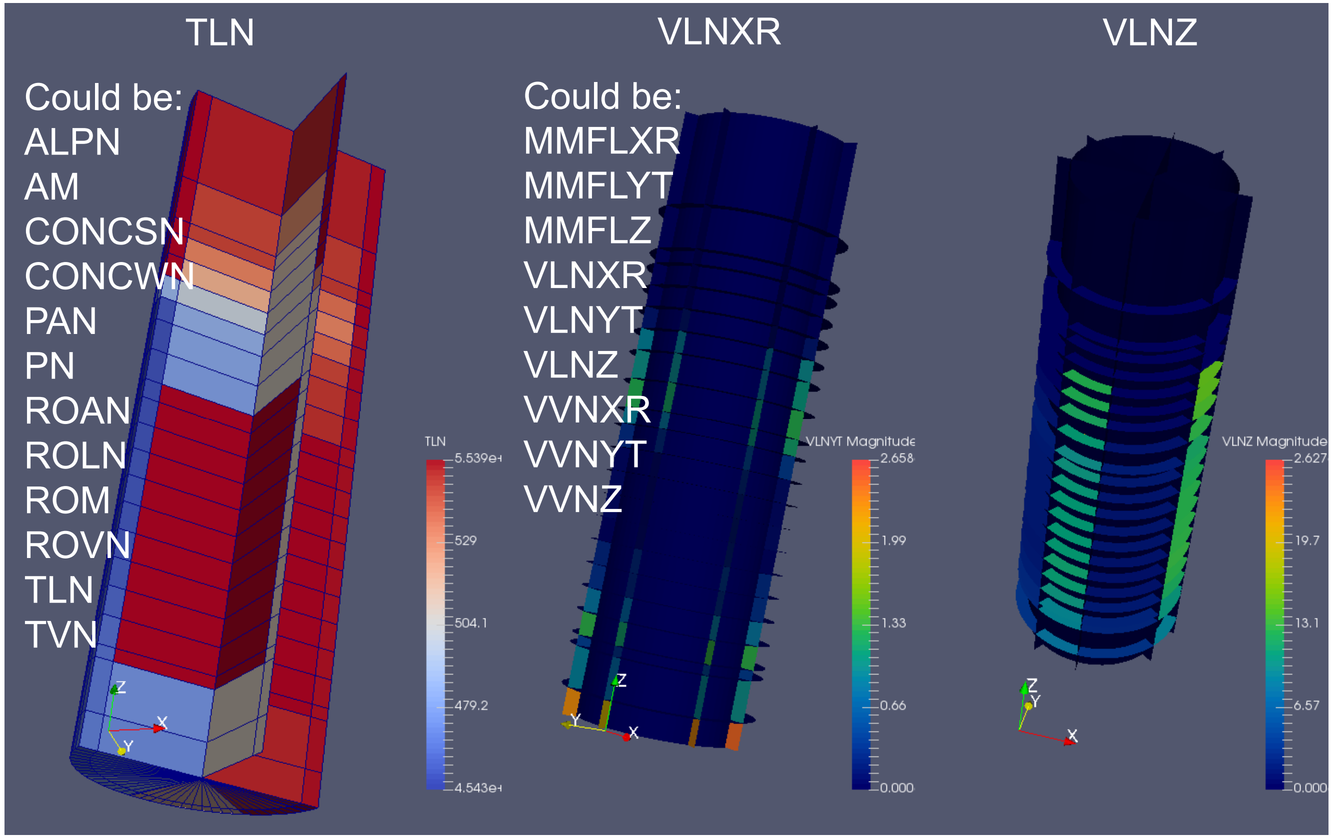
Temperature (inlet 1, 3) = 453.7 K

Temperature (inlet 2, 4) = 553.7 K

Additional features for the vessel coupling cases

Karlsruhe Institute of Technology

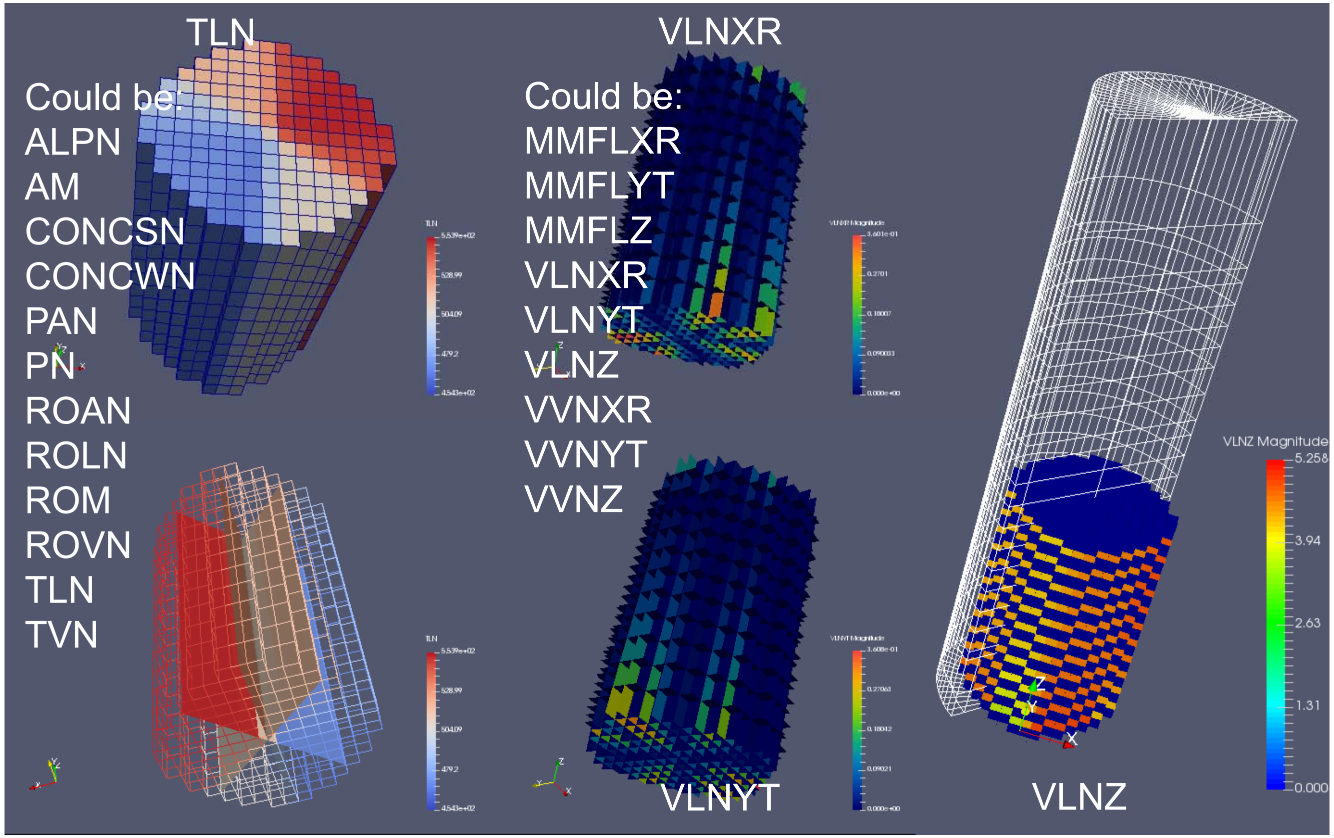
2) Post-processing for the Cylinder Vessel and Cartesian Vessel



Additional features for the vessel coupling cases

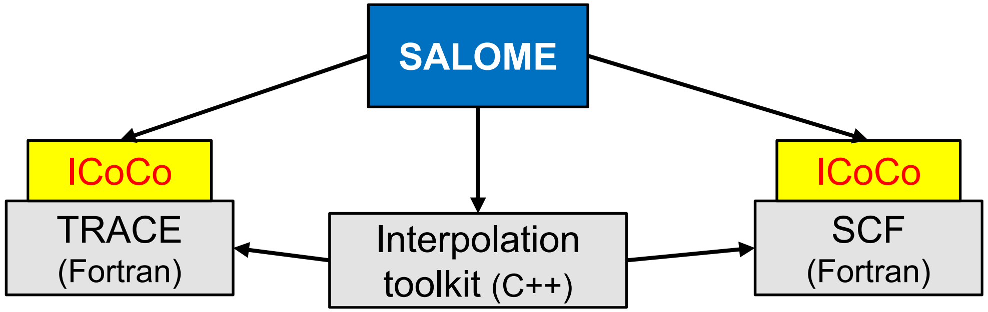
Karlsruhe Institute of Technology

2) Post-processing for the Cylinder Vessel and Cartesian Vessel



Summary

1) **TRACE and SCF are coupled through ICoCo in SALOME.**



2) **The coupling of Cylinder and Cartesian Vessels is fully automated.**

   a) Users only need to define the Vessels for vessel coupling cases, the inter-junctons between vessels will be automatically established and the flow area will be automatically calculated.

   b) Volumetric and face meshes are constructed for both vessels and various fields are stored in the meshes for post-processing.

   c) Users can decide to compile classic TRACE or TRACE with auto-coupling and post-processing functions by identify a compiling flag.

Thank you for your attention.