OPENSC²: the thermal-hydraulic modelling of SuperConducting Cables moves towards OPEN science

Laura Savoldi, Daniele Placido, Sofia Viarengo

MAHTEP Group, Department of Energy "Galileo Ferraris", Politecnico di Torino, Italy

CHATS-AS, 23/09/2021

Outline

- Framework and aim of the work
- Test-Driven Development: 3 case studies
- OPENSC^2: a novel object-oriented tool
- Case studies:
  - ITER TF cable
  - HTS cable for fusion
  - HTS cable for power transmission
- OPENSC^2 repository and collaborative incremental development
- Conclusions and next steps

Framework: tools for TH modelling of fusion cables

Feature	THEA/SUPERMAGNET	VINCENTA/VENICIA	M&M/4C
Use	Commercial license	Commercial license	Proprietary
Geometry	Arbitrary	Wide range of cases	CICCs with ≤3 channels; some HTS cables (H4C)
Fluids	SHe	He, N_2, O_2, Ne, H_2O	SHe
LTS	Y	Y	Y
HTS	Y	?	Y (H4C)
Spatial discretization	FEM (adaptive)	3rd order FDM	1st order FEM (adaptive)
Time integration scheme	3rd order adaptive multi-step	Semi-explicit splitting-up	Adaptive BE or CN
Pre and post-processing	inner post-processor	Inner pre- and post-processor	Performed with external tools
GUI	Y	Y	N
Validation	++	+	+++

State of the art of TH modelling for SC cables for power transmission →

Open Science and aim of the work

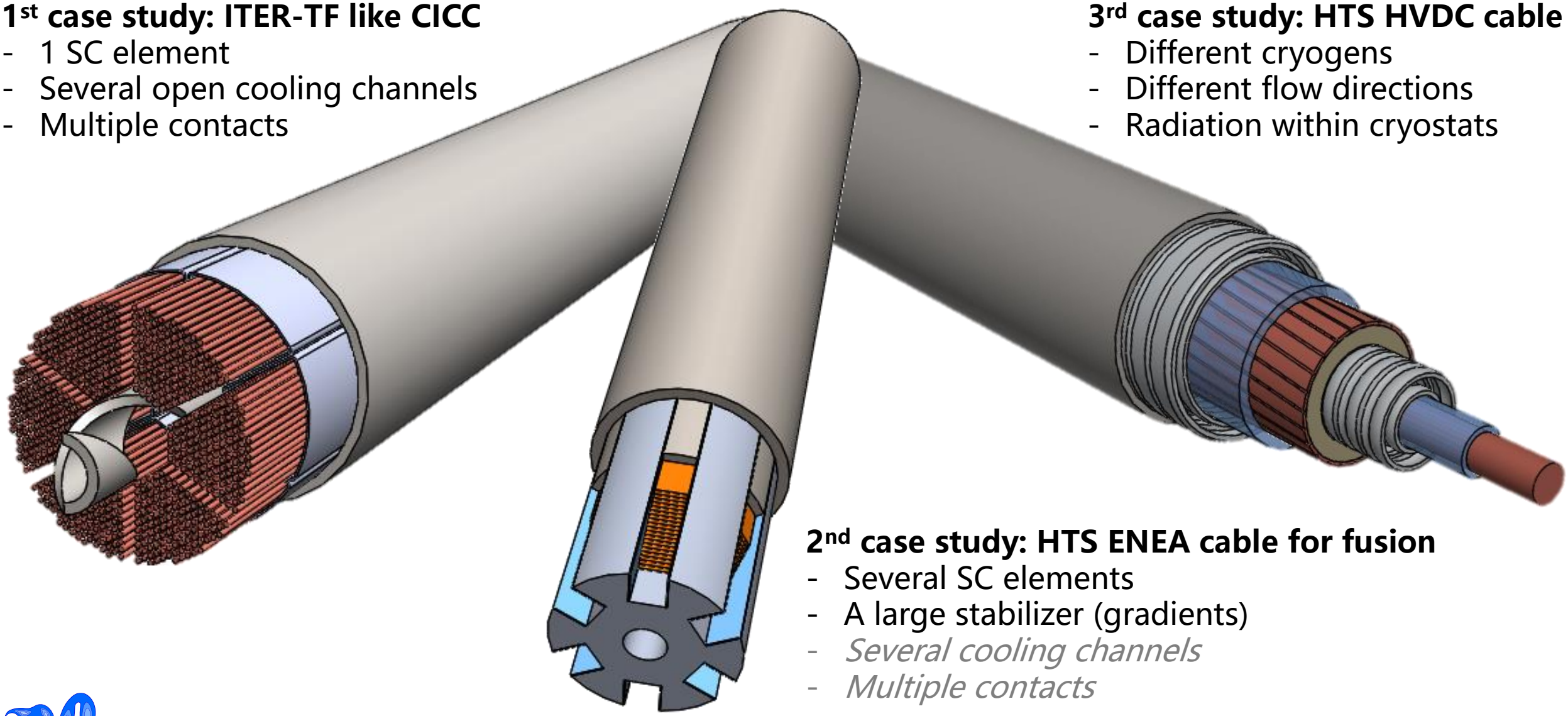
- Twofold motivation to move to Open Science:

- Fusion projects as public endeavors
- AIM = establish a novel cooperative framework for an open-source tool for the numerical modelling of TH transients in SC cablesc, bridging the gap between established tools and a new generation of researchers.

Test-Driven Development: case studies

1st case study: ITER-TF like CICC
- 1 SC element
- Several open cooling channels
- Multiple contacts

3rd case study: HTS HVDC cable
- Different cryogens
- Different flow directions
- Radiation within cryostats



2nd case study: HTS ENEA cable for fusion
- Several SC elements
- A large stabilizer (gradients)
- *Several cooling channels*
- *Multiple contacts*

OPENSC²: a novel object-oriented tool

- Object-oriented approach implemented in Python

CLASS
Attributes…
Methods… Methods…

*Each conductor object **CO** from class CONDUCTOR*

CLASS: CONDUCTOR
Length **Structure** Connectors among components Operational attributes Operating scenario
Initialization of components Solution of transient Post-processing

CLASS: SOLIDS
Material **Geometrical attributes**
Transient 1D energy conservation law

LIBRARY SOLID MEDIA (open database)

***SOs** and **JOs** are "children" (instances) of the class SOLIDS*

OPENSC²: a novel object-oriented tool

- Object-oriented approach implemented in Python

CLASS
Attributes…
Methods… Methods…

*Each conductor object **CO** from class CONDUCTOR*

CLASS: CONDUCTOR
Length **Structure** **Connectors among components** Operational attributes Operating scenario
Initialization of TH components Solution of transient Post-processing

CLASS: FLUIDS
Material **Geometrical attributes**
1D Mass, momentum and energy conservation laws

LIBRARY FLUID MEDIA (Coolprop)

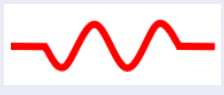
Fluid Objects
FOs

FOs are istances of the class FLUIDS

*The tool is intrinsically multi–conductor: several **CO**s can be instantiated simultaneously*

OPENSC²: connectors between objects

→ *Each **CO** is assembled using SOs, FOs and Jos, connected through connectors*

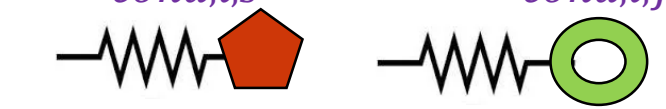
Object	Interaction	Symbol	Object
SO	Thermal by conduction		SO
SO	Thermal by conduction		JO
JO	Thermal by conduction		JO
JO	Thermal by radiation		JO
SO	Thermal by convection		JO
JO	Thermal by convection		FO
FO	Thermal through an impermeable surface		FO
FO	Hydraulic and thermal by a permeable surface		FO

OPENSC²: models for the SOs and JOs

- Energy equation: 1D transient heat conduction equations with sources

SOs

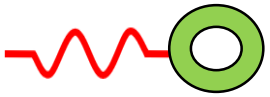
$$A_i\rho_i c_{p,i}\frac{\partial T_i}{\partial t} - A_i\frac{\partial}{\partial x}\left(k_i\frac{\partial T_i}{\partial x}\right) = Q_{operation}(t,x) + \sum_{s\neq i}^{N_S}\frac{(T_s-T_i)}{R_{cond,i,s}} + \sum_{j=1}^{N_J}\frac{(T_j-T_i)}{R_{cond,i,j}} + \sum_{f=1}^{N_F}\frac{(T_f-T_i)}{R_{conv,i,f}}$$



AND/OR

JOs

$$A_i\rho_i c_{p,i}\frac{\partial T_i}{\partial t} - A_i\frac{\partial}{\partial x}\left(k_i\frac{\partial T_i}{\partial x}\right) = Q_{environment}(t,x) + Q_{operation}(t,x) + \sum_{s=1}^{N_S}\frac{(T_s-T_i)}{R_{cond,i,s}} + \sum_{j\neq i}^{N_J}\frac{(T_j-T_i)}{R_{cond,i,j}} + \sum_{j\neq i}^{N_J}\frac{(T_j-T_i)}{R_{rad,i,j}} + \sum_{f=1}^{N_F}\frac{(T_f-T_i)}{R_{conv,i,f}}$$

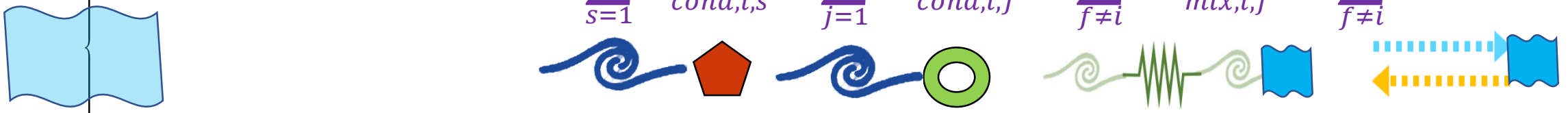


OPENSC²: models for the FOs

- For <u>supercritical/gaseous media</u>: 1D mass/momentum/energy conservation laws for an inviscid fluid, rewritten in for the variables *v, ρ, T*

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} = \frac{1}{\rho}\Lambda_{v+\rho}$$

$$\frac{\partial p}{\partial t} + \rho c_s{}^2\frac{\partial v}{\partial x} + v\frac{\partial p}{\partial x} = \frac{\Phi}{A}\left[F + \sum_{s=1}^{N_S}\frac{\left(T_s - T\right)}{R_{cond,i,s}} + \sum_{j=1}^{N_J}\frac{\left(T_j - T\right)}{R_{cond,i,j}} + \sum_{f\neq i}^{N_F}\frac{\left(T_f - T_i\right)}{R_{mix,i,f}} + \sum_{f\neq i}^{N_F}\Lambda_{\rho+v+e}\right]$$



$$\frac{\partial T}{\partial t} + \Phi T\frac{\partial v}{\partial x} + v\frac{\partial T}{\partial x} = \frac{1}{A\rho c_v}\left[F + \sum_{s=1}^{N_S}\frac{\left(T_s - T\right)}{R_{cond,i,s}} + \sum_{j=1}^{N_J}\frac{\left(T_j - T\right)}{R_{cond,i,j}} + \sum_{f\neq i}^{N_F}\frac{\left(T_f - T_i\right)}{R_{mix,i,f}} + \sum_{f\neq i}^{N_F}\Lambda_{\rho'+v+e}\right]$$

Different models for liquid or 2-phase flow still to be implemented

OPENSC²: numerics

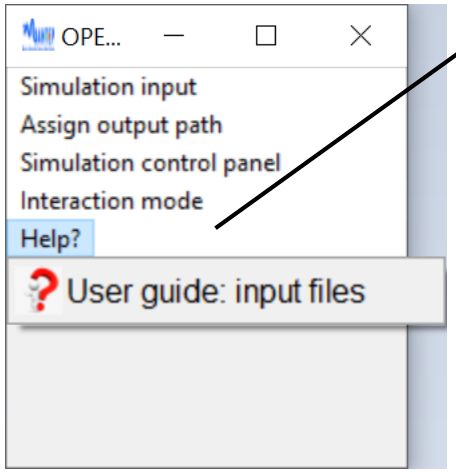
For the time being:

- **SOs, JOs**: Conservative recipe for the heat capacity of material mixtures
- **SOs, JOs** & **FOs**: uniform or non-uniform static grids
  - Coefficients computed at the Gauss point
  - P_1 FE in space for all objects.
- **COs**: 1-step integration schemes for time marching, constant time stepping

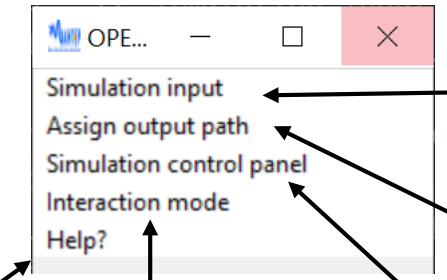
- Boundary conditions:
  - **SOs, JOs**: adiabatic at both boundaries
  - **FOs**: different possible sets for subsonic flow:
    - Impose p_{in}, T_{in}, p_{out}
    - Impose v_{in}, T_{in}, p_{out}(~inlet initial mass flow rate and inlet temperature)
    - …

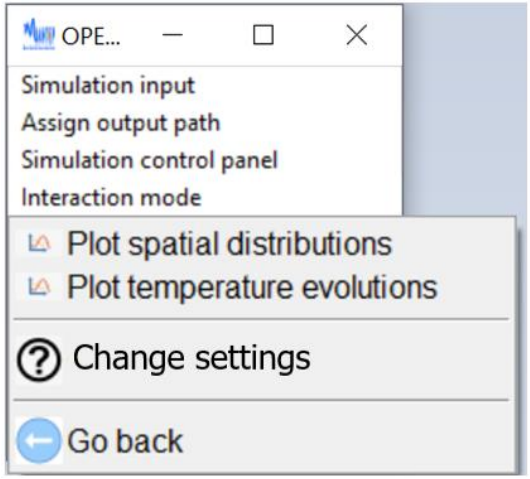
OPENSC²: (graphical) user interface

User-friendly GUI to control simulation **features** and **behaviour**

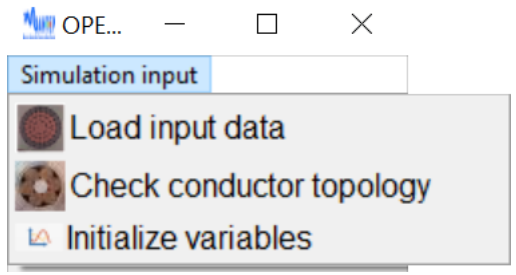


Interactive User Guide for compiling Input files

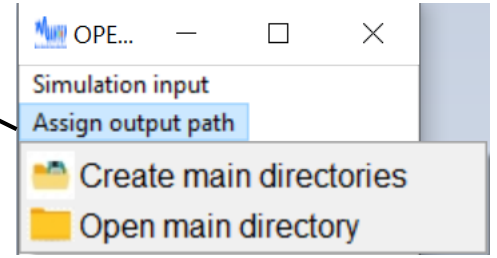




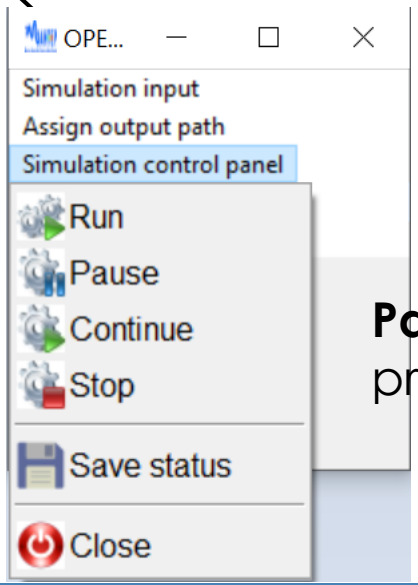
RUN-AND-CHECK (*development ongoing*)



Load input XLS files

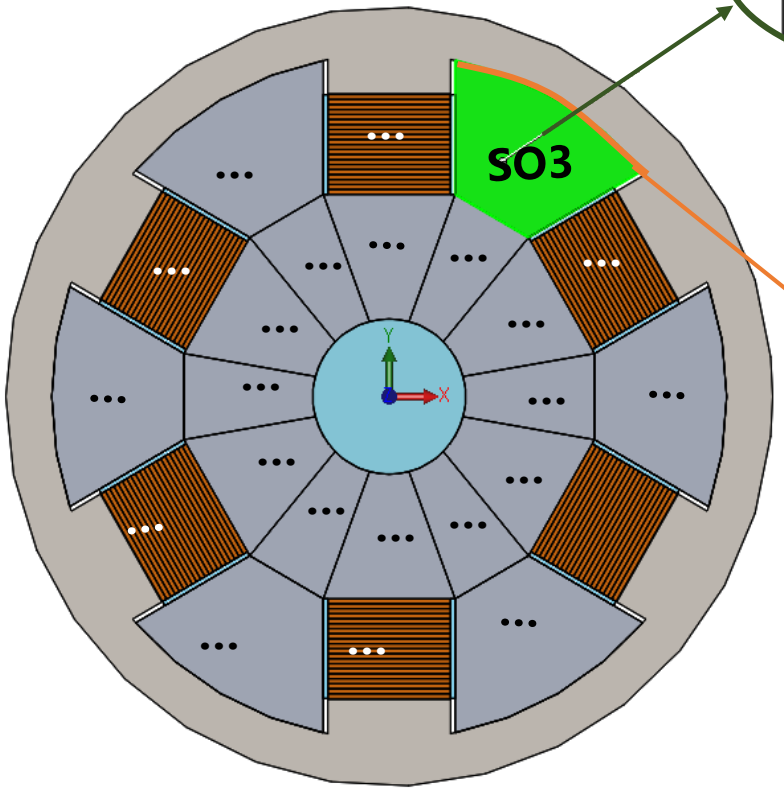


Create / open directory to store the simulation output



Pause / Restart simulation provided by user

OPENSC2: INPUT



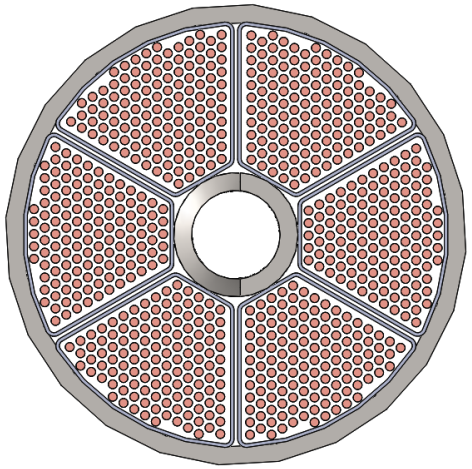
Area:	21.0568mm^2
Perimetro:	18.5552mm

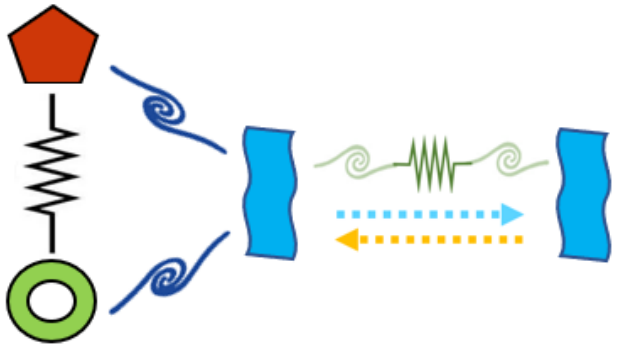
	A	E	F	G
3	**Variable name**	**SO1**	**...**	**SO3**
4	CROSSECTION	...	...	21.0568×10^{-6}

	A	Y	AM	AN
1		**Contact perimeter for objects coupling**		
2		**SO4**	**SO18**	**JO1**
24	**SO2**	0	0	0
25	**SO3**	0	0	7.191×10^{-3}

Lunghezza arco: 7.1913mm
Lunghezza corda: 7.0639mm
Raggio: 11.0000mm
Angolo: 37.46gradi

Case study 1: an ITER TF conductor (1)



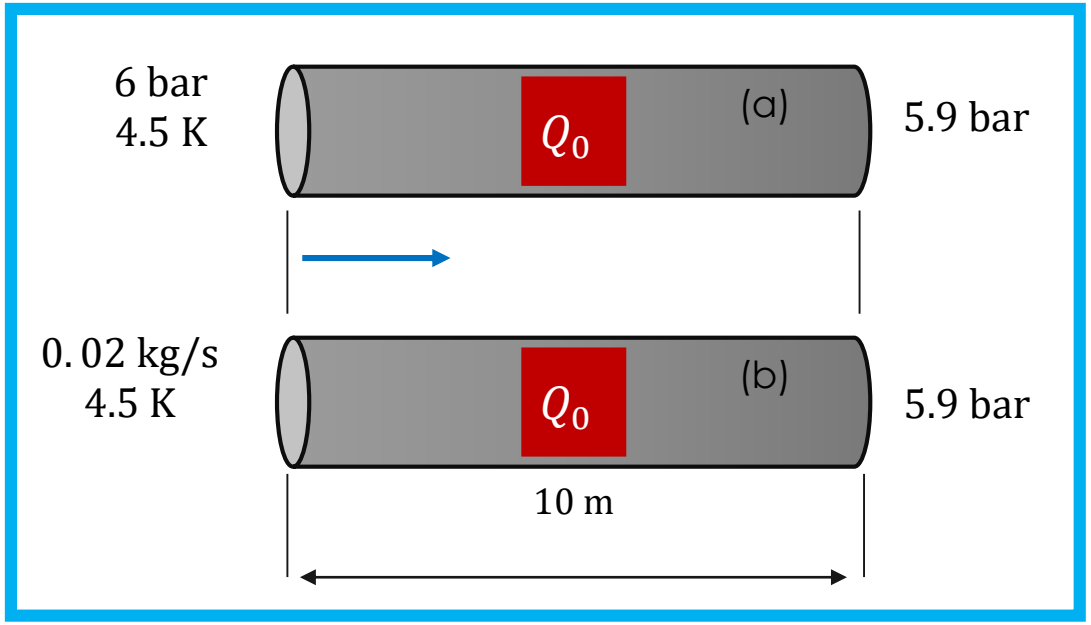


- Heat the central 2 m of the strands with 250 W/m for 10 s .
- Different sets of BCs

- Instance of the Conductor class assembled as:

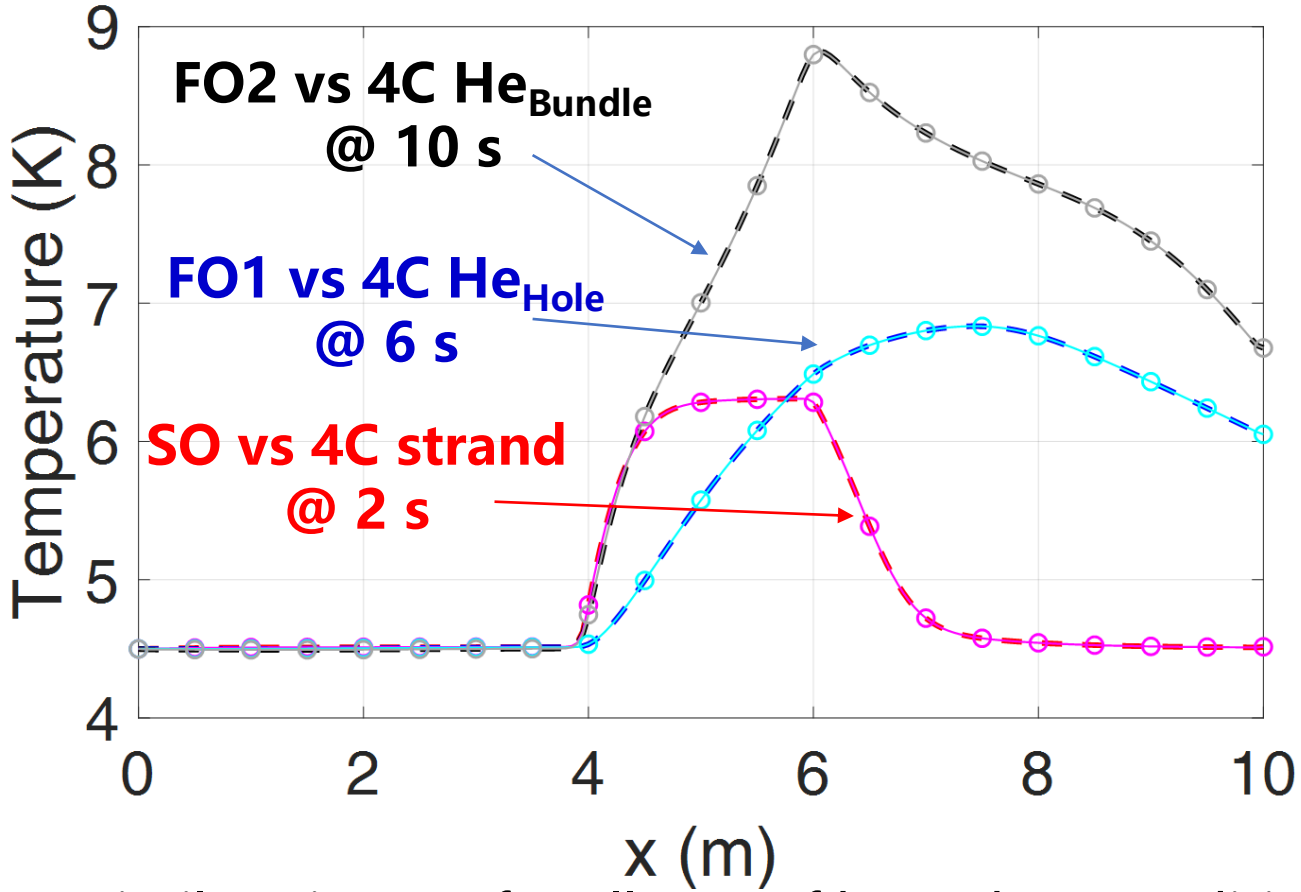
  CO = 1 **SO** + 1 **JO** + 2 **FO**s

- Consider a 10m-long conductor

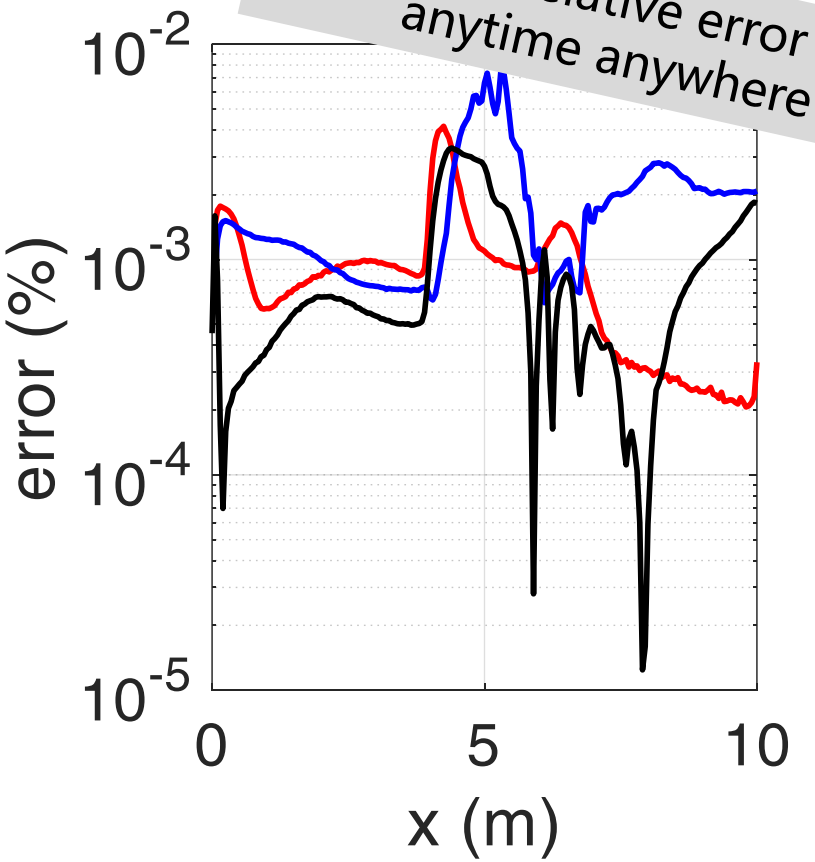


Case study 1: benchmark with the 4C code

- Test @ **Constant inlet velocity,** space and time convergence checked (1st order here)
- BENCHMARK with the 4C code

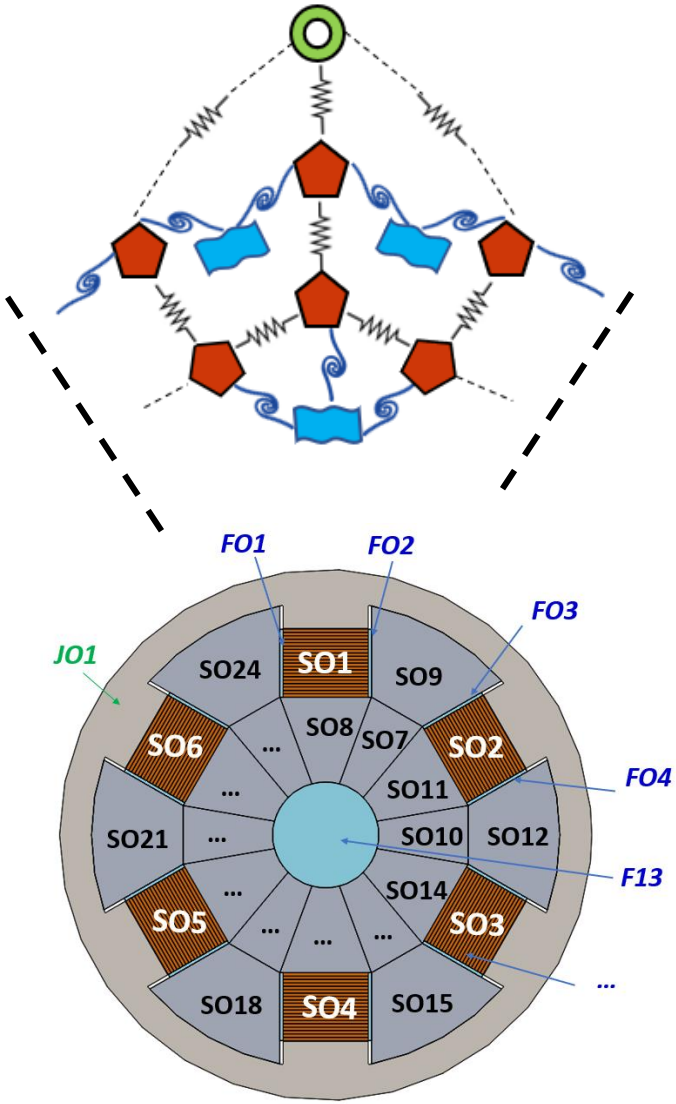


Maximum relative error < 10^{-4} anytime anywhere



- Similar pictures for all sets of boundary conditions / drivers tested so far

Case study 2: an HTS cable for fusion

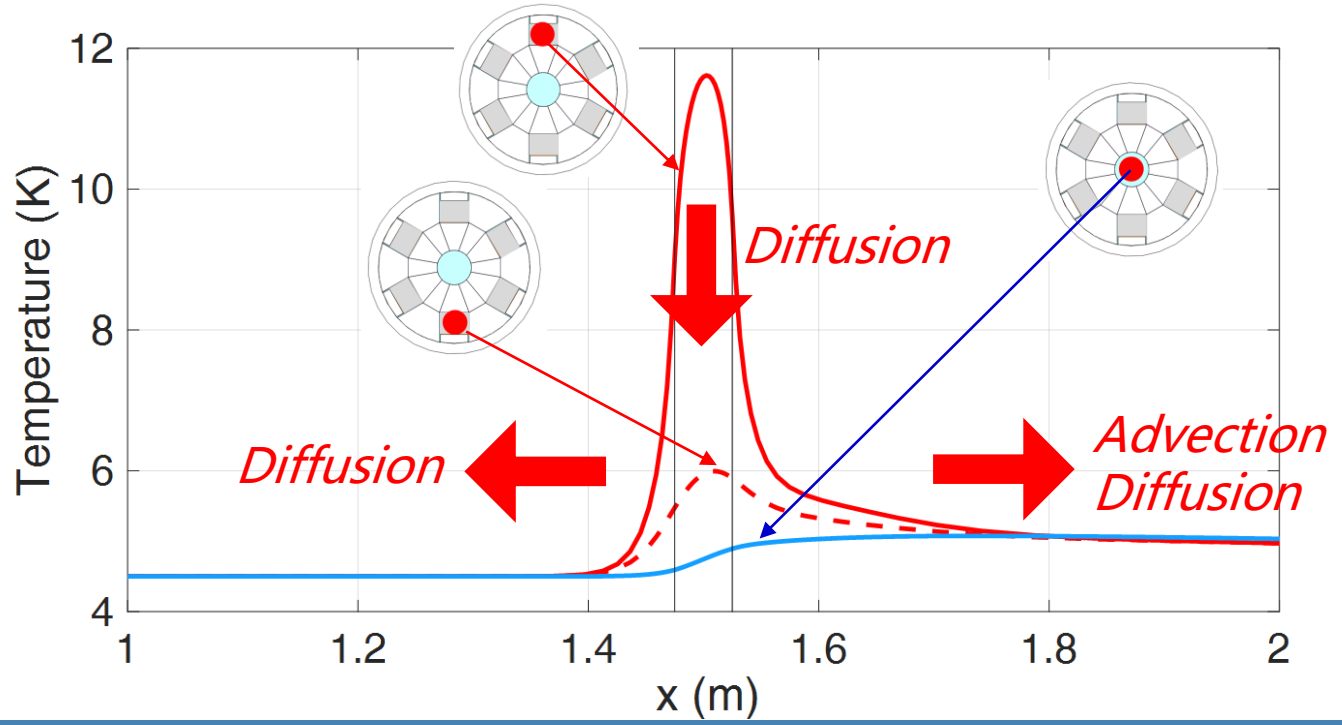


- Instance of the Conductor class assembled as:

  CO = (18+6) **SO** + 1 **JO** + 13 **FO**s

  [A. Zappatore at al, IEEE TAS 2020]

- Consider a SULTAN-like 3m-long conductor, cooled by 5 g/s of SHe @ 4.5 K, 10 bar; heat the central 5 cm of SO1 with 250 W/m for 1 s

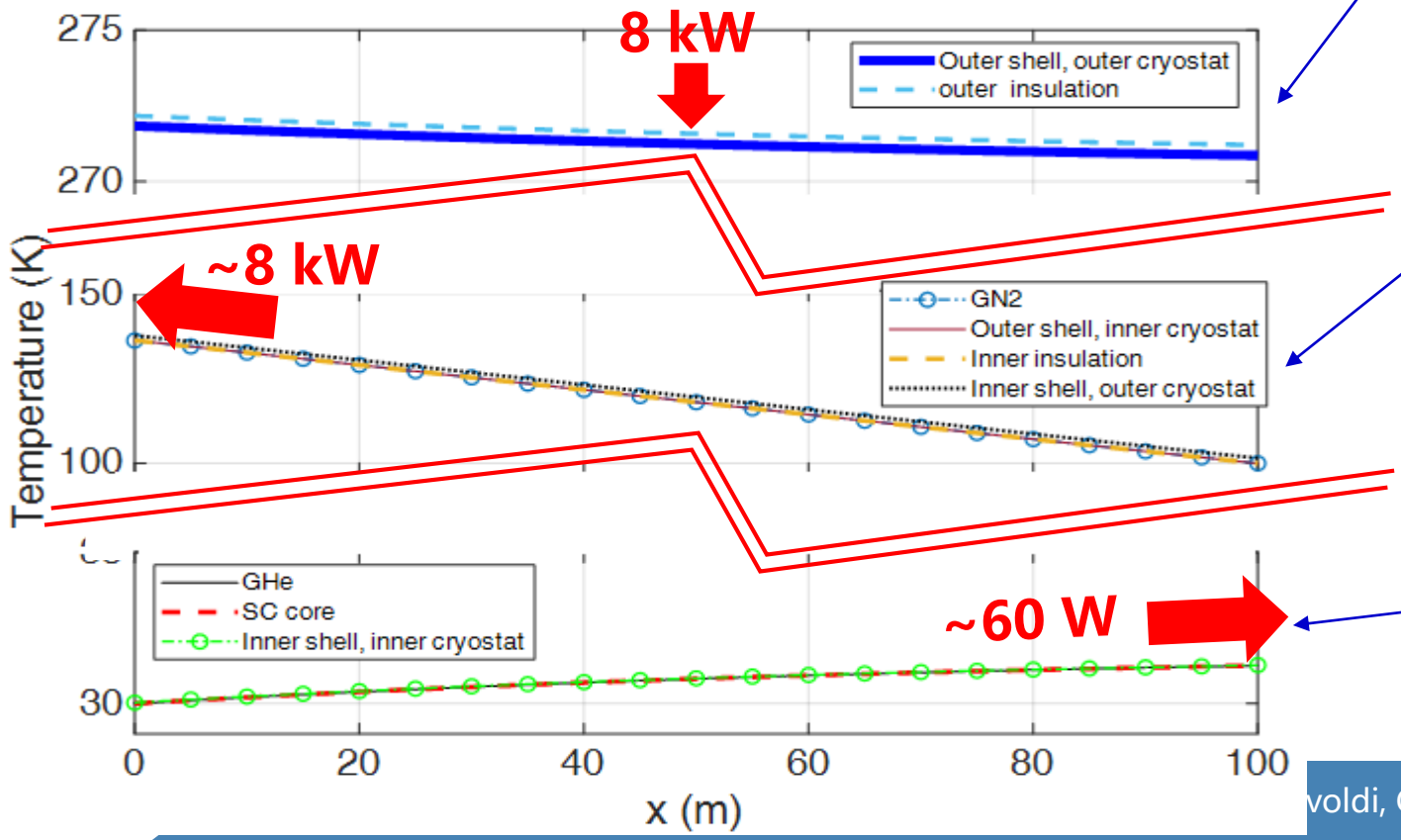


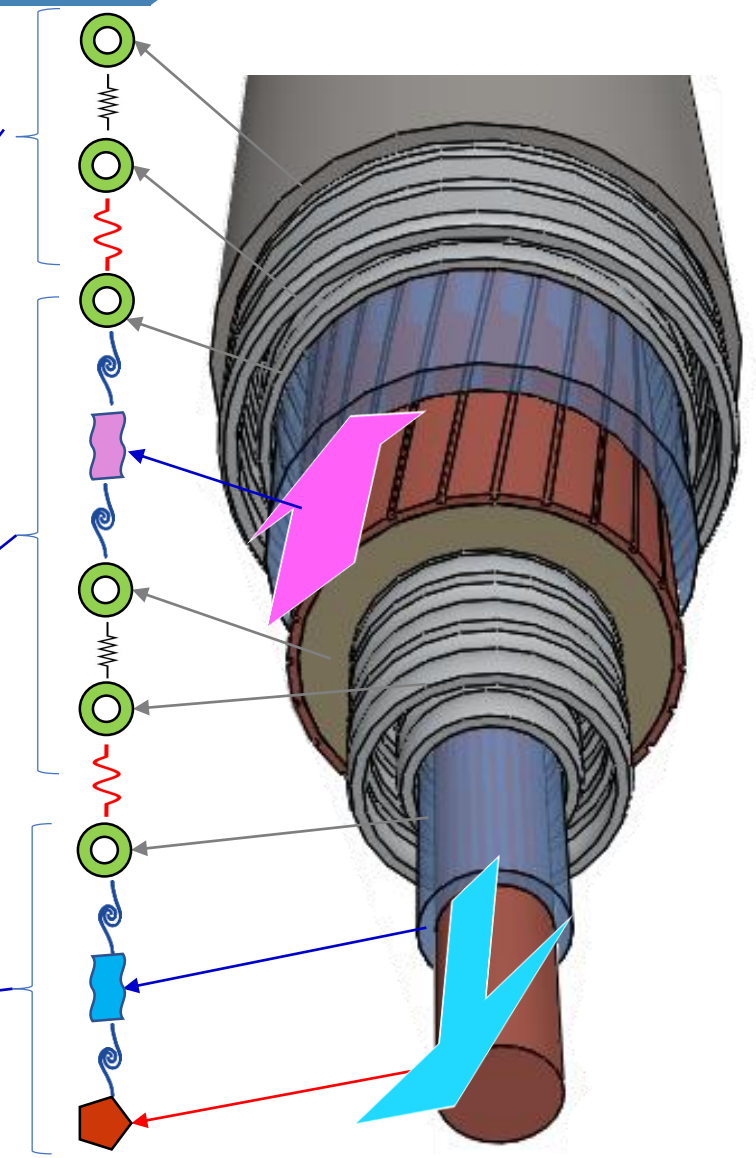
Case study 3: an HTS cable for power transmission

- Instance of the Conductor class assembled as:

  CO = 1 **SO** + 7 **JO**s + 2 **FO**s **1 GHe**, **1 GN_2 in counter flow**

- Operating condition under parasitic load from the environment
- ~10 g/s of He @ 30K, ~ 0.20 kg/s of N_2 @ 100 K





OPENSC² repository and collaborative incremental development

- Developed in Python version 3.8.6 → inherits the licence of Python and its packages
- Distribution *copyleft* () on GitHub (*)
- Call to the TH modelers community
  - To carry on benchmarks against their own tools
  - To give feedbacks to the developer team @ PoliTo
  - To identify new needs
  - To contribute to the development of new features

Conclusions and next steps

- Object-oriented open tool for the TH analysis of fusion and power SC cables available soon for researchers, with several functionalities already implemented
- Test-Driven Development allows successful simulation of different test cases (benchmarked when possible)
- Framework for collaborative development in place

- Adaptive grid and time stepping under test
- Coupling to a current diffusion model (resistive-inductive network) coming at MT27
- Other models for different fluid media under development

- GUI further improvement, including possible coupling with FreeCAD